MULTIPLE UAV SIMULATION WITH MULTIRESOLUTION MULTISTAGE MODELS AND DECISION SUPPORT

Except where reference is made to the work of others, the work described in this thesis is my own or was done in collaboration with my advisory committee. This thesis does not include proprietary or classified information.

Bo Li

Certificate of Approval:

Min-Te Sun
Assistant Professor
Computer Science and
Software Engineering

Alvin Lim, Chair
Associate Professor
Computer Science and
Software Engineering

Cheryl Seals
Assistant Professor
Computer Science and
Software Engineering

George T. Flowers
Dean
Graduate School

MULTIPLE UAV SIMULATION WITH MULTIRESOLUTION MULTISTAGE MODELS AND DECISION SUPPORT

Bo Li

A Thesis

Submitted to

the Graduate Faculty of

Auburn University

in Partial Fulfillment of

the Requirements for the

Degree of

Master of Science

Auburn, Alabama
August 9, 2008

MULTIPLE UAV SIMULATION WITH MULTIRESOLUTION MULTISTAGE MODELS

AND DECISION SUPPORT

Bo Li

____

Signature of Author

____

Date of Graduation

VITA

Bo Li, son of Zhengyi Li and Xiaoli Bao, was born on June 1, 1979 in Beijing, P. R. China. He graduated from Beijing No. 22 Middle School in 1995. He entered Jin Lin University, Changchun, P. R. China and graduated *summa cum laude* with a Bachelor of Science degree in Computer Science in July, 2002. From January 2003 to August 2003, he worked at Beijing CTNA Technology CO., Ltd as a software programmer. From August 2003 to November 2005, he worked at Panasonic Corporation of China as a system administrator and data analysis specialist. After that, he entered Graduate School of Auburn University majoring in Computer Science in January, 2006. During his graduate study at Auburn University, he performed well on both his academic study and research work. From January 2006 to July 2006, he was involved in part time work at Distance Learning Department of Auburn University as a programmer. From 2006 to 2007, he was a Research Assistant, and from 2006 to 2008 he was a Teaching Assistant in the Computer Science Department of Auburn University.

THESIS ABSTRACT

MULTIPLE UAV SIMULATION WITH MULTIRESOLUTION MULTISTAGE MODELS AND DECISION SUPPORT

Bo Li

Master of Science, August 9, 2008
(Bachelor of Science, Ji Lin University, 2002)

109 Typed Pages

Directed by Alvin Lim

Multi UAV simulation is a dynamic multi-simulation of multiple Unmanned Aerial Vehicles (UAV) activities in the battlefield with multiresolution multistage models. Decision support is implemented under this simulation. It provides an intelligent agent supported by decision making subsystem based on the decision making model. It deals with the uncertainty of the battlefield and presents a multiple simulations with Multiresolution Multistage (MRMSM) model. Unmanned Aerial Vehicle is one of the effective devices in the future battlefield and has many advantages. Single UAV has been used in the battlefield. Multiple UAVs have more advantages than a single UAV to attack a group of targets. A UAV team can accomplish tasks that a single one can not execute alone. The requirement to coordinate multi-UAVs continues to surface in both defense and civil applications. Multiple UAVs form teams and need to cooperate well in the battlefield to attack efficiently and effectively. Decision support subsystem generates two strategies that could be selected for the UAV team action. Bayesian Network is used for making decisions to select a strategy for UAV team in a specific situation. Based on the selected strategy, UAV team could attack the targets in an effective manner. In this simulation, there are multiple simulations working on different platforms at multiple stages. They exchange information in a timely manner with each other and are dynamically launched, updated or reconfigured, and stopped. These actions are supported by the decision making model. There are five main parts in this simulation, Matlab Simulink, Filter, Tactical Federate, High resolution and Visualization. Each part works on its own platform. These parts dynamically work with each other and communicate with each other and exchange data timely. This paper focuses on the Simulink, Filter, Tactical federate and communication between parties. The results show that each part can work independently and cooperate well with each other. UAV team can effectively attack battalions which are target clusters. The two-dimension visualization for low resolution and three-dimension visualization for high resolution show the entire simulation scenario at different stages.

ACKNOWLEGEMENTS

I would like to express special gratitude to my advisor Dr. Alvin Lim, professor, Department of Computer Science and Software Engineering at Auburn University, for his instruction, guidance, encouragement and patience in completion of the research and thesis. In particular, his suggestions, criticisms and general exchange of ideas greatly contributed to this thesis.

Thanks, also, due to his advisory committee members, Dr. Yilmaz, Dr. Seals and Dr.Sun and the professors and staff members in the Department of Computer Science and Software Engineering at Auburn University for their kindness and help through these years.

Finally, a sincere thank to my friend Haihong Liang. She gave her greatest support and encouragement to help me finish all the research work. Also, I thank my parents, who provide the inspiration, encouragement and support to me during these years.

Style manual or journal used: Guide to Preparation and Submission of Thesis and Dissertation

Computer software used: Microsoft Word 2003, PowerPoint 2003

TABLE OF CONTENTS

LIST OF FIGURES xi

CHAPTER 1: INTRODUCTION AND BACKGROUND 13

1.1 General Introduction 13
1.2 Multiresolution Multistage Model (MRMSM) 14
1.3 Multisimulation 15
1.4 Multisimulation with MRMS Model 16
1.5 Decision Making 16
1.6 Situation Awareness 18
1.7 Integration Mulisimulation with Decision Making Support Subsystem 18
1.8 Unmanned Aerial Vehicle 18
1.9 Matlab Simulink 20
1.10 Filter 21
1.10.1 Introduction 21
1.10.2 Contract Net protocol 22
1.10.3 Cluster Algorithm 22
1.11 Tactical Federate 23
1.12 High Resolution 25
1.12.1 Introduction 25
1.12.2 VR-Link, HLA Mak RTI, Stealth 26
1.13 Strategy for UAV Team 27

CHAPTER 2: MOTIVATIONS 33

CHAPTOR 3: OBJECTIVES 35

CHAPTER 4: CHALLENGING PROBLEMS 37

4.1 The Problem of Deep Uncertainty 37
4.2 Multiple Simulations at Multiple Stages 37
4.3 UAV Team Work with Strategy Support and Targets Clustering 38

CHAPTER 5: RELATED WORKS 40

5.1 Distributed Task Allocation 40
5.2 Path Planning 42
5.3 Consistency Maintenance 43
5.4 Multi-agents System 44

5.5 Multi-agents Market-based System 46
5.6 Genetic Algorithm....47
5.7 Evidential Map-building....48
5.8 Team Work....49
CHAPTER 6: GENERAL APPROACHES 52
6.1 Introduction....52
6.2 Simulation Scenario....52
6.3 Low Resolution....54
6.4 Filter....56
6.5 Tactical Federate 56
6.6 High Resolutions....58
CHAPTER 7: DETAILS OF DESIGN AND SIMULATION.... 61
7.1 Introduction....61
7.2 Low Resolution....62
7.2.1 Initialization Model....63
7.2.2 Communication Model 63
7.3 Filter....65
7.3.1 Introduction....65
7.3.2 Targets Clustering 65
7.3.3 Team Formation 66
7.3.4 Consensus Model....67
7.3.5 Communication Functions of Filter....68
7.4 Tactical Federate 69
7.4.1 Strategy Selection 69
7.4.2 Communication Functions of Tactical Federate 74
CHAPTER 8: SIMULATION RESULTS COMPARISON 95
CHAPTER 9: SUMMARY AND FUTURE WORKS.... 102
REFERENCES.... 104

LIST OF FIGURES

Figure 1.1 Unmanned Aerial Vehicles (UAV). Predator 28

Figure 1.2 Multiresolution Model 29

Figure 1.3 Mulitstage Model 29

Figure 1.4 Simulation Updating and Branching for Multisimulation 30

Figure 1.5 Matlab Simulink 31

Figure 1.6 Formation Flight of UAV Team and Flock of Birds (Song et al., 2005) 32

Figure 6.1 Low Resolution Visualization 59

Figure 6.2 High Resolution Visualization 60

Figure 7.1 Low Resolution at Initialization Stage 75

Figure 7.2 “Save Plot Data” Block in Simulink 76

Figure 7.3 “Save Plot Data” inside Structure 77

Figure 7.4 Time Unit of the S-Function is set to 0.1 second 78

Figure 7.5 Simulink and Filter Communication Sequence Chart 79

Figure 7.6 Communication between Simulink and Tactical Federate 80

Figure 7.7 A Simple Targets Clustering Process 81

Figure 7.8 Targets are Clustered as Battalions 82

Figure 7.9 Team Formation Process Based on Contract Net 83

Figure 7.10 Team Formation Process Results 84

Figure 7.11 Filter and Tactical Federate Communication Sequence Chart 85

Figure 7.12 Communication between Filter and Tactical Federate 86

Figure 7.13 MSBNx Implementation 87

Figure 7.14 Sample of Critical Target 88

Figure 7.15 Assessment Table of “Number of Target” Node 89

Figure 7.16 Assessment Table of DSC Node 90

Figure 7.17 Calculation Results of Bayesian Network 91
Figure 7.18 Simulation using Fringe Point strategy (1) - Fringe and Fringe Points of Targets Area 92
Figure 7.19 Simulation using Fringe Point Strategy (2) 92
Figure 7.20 Simulation using Fringe Point Strategy (3) 93
Figure 7.21 Simulation using Region Strategy 93
Figure 7.22 Program Sequence Chart of Tactical Federate Communication Functions 94
Figure 8.1 UAV Destroy Target by Time Lapse 99
Figure 8.2 Target Destroy Rate by Using Different Strategies 100
Figure 8.3 Efficiency of UAV Destroy Target Using the Strategies 101

CHAPTER 1: INTRODUCTION AND BACKGROUND

1.1 General Introduction

Unmanned Aerial Vehicles (UAV) (see Figure 1.1) is widely used in today's battle field. Simulating multiple UAV team to attack a group of targets effectively is a problem. The goal of this simulation is to build a decision making system based on exploratory multisimulation technology integrated to the subsystem based on Naturalistic Decision Making theory. We simulate situation awareness and exploratory multisimulation to evaluate the efficiency and effectiveness of multiple actions at different levels of resolution.

UAVs search and sense the area, perceive the situation, form teams and attack battalions based on strategies. The ability to use cooperative autonomous vehicles to perform a wartime mission is an important application in future military operations (Brown 1998, Davis 1995, McHale 2003, French 2003) [1]. This simulation focuses on UAV team work. There are multiple simulations and models in this simulation. They work respectively and cooperatively. Some simulations work on different platforms. They communicate with each other in a timely manner to exchange information and keep consistence between each other. Under the decision making model, there are multiple simulations with multiple stages. There are 2D visualization for low resolution and 3D visualization for high resolution. These visualizations are consistent with each other.

1.2 Multiresolution Multistage Model (MRMSM)

Resolution is the degree of detail and precision used in the representation of real world aspects in a model or simulation [2]. Multi-Resolution Modeling involves the following: [3]

(1) Building a single model with alternative user modes involving different levels of resolution for the same phenomena;

(2) Building an integrated family of two or more mutually consistent models of the same phenomena at different levels of resolution; or

(3) Both.

There are two resolutions in this simulation, low resolution and high resolution. Since low resolution is more abstract than high resolution, it is at a high level and high resolution is at a low level. Multiresolution model (see Figure 1.2) composes multiple single models, or same model at different levels of resolution. The Multiresolution Multistage Model integrates multiresolution models with multistage models (see Figure 1.3) to help deal with the problem of multiple tasks that are at different resolution levels. Simulation entities need to be capable of simultaneously operating at different levels and maintain consistency at each level. Federates need to switch between the submodels. Models of entities are needed to be dynamically updated across the resolutions at run-time. Low resolution should exchange results with high resolution. Models have different stages and can switch as the resolution level of changes. Decision making subsystem need to make its decision and inject its judgments to the multiple subsystems. Since the uncertainty of the battlefield scenario, strategy decision will be made every period of time and dynamically update the simulations which may make change of the resolution level. Figure 1.2 is Multiresolution model. There are three models, Model 1 (M1), Model 2 and Model 3 representing same entity at different levels of resolution. Figure 1.3 is the Multistage model in which there are two subset models. Only one subset model is active at one stage unless there are multiple simulations. The controller decides when and how the subset model switches from one to another. If the sub-model in one stage constitute of model of the same entity at different levels of resolution, the model is a Multiresolution Multistage model (MRMSM) [4], i.e., the UAV entity is at both low and high resolution at the same stage such as attacking.

1.3 Multisimulation

Multisimulation (or multisim, for short) is simulation of several aspects of reality in a study. It includes simulation with multiresolution model, multiaspect model and multistage model. In multisimulation model, decision supports model updating or changing. The associated parameters and experimental conditions are used in the subsequent stages of simulation [4]. In this simulation, a new model or simulation can be invoked at the interruption point. This work is based on the assessment of the previous simulation to deal with the uncertainty of the scenario. In each stage, there are multiple simulations work on different levels and cooperate with each other. Models and different simulations can be invoked or stopped during each stage. Multisimulation updating process is showed as Figure 1.4.

1.4 Multisimulation with MRMS Model

Different resolutions work independently and cooperatively such as low resolution and high resolution. There are many Course-of-Actions (COA) for a resolution. In this simulation, multiple high resolutions are implemented to analyze and assess COAs. The model and the components of the models are dynamically updated based on demands and assessment of previous work during the simulation. The UAV team in high resolutions can perform based on different strategies generated by low resolution strategy decision model which is in Tactical Federate. The multiple resolutions or simulations run on different machines or simulation platforms in parallel and corporation work. MRMS Models need to dynamically and timely exchange information with each other to make consistency between low resolution and high resolution. It needs run-time enforcement policy for multiple resolutions consistency when the models update. It also needs to flexibly update resolution levels on demand and change the behavioral models of each entity at run-time and at different resolution levels [4].

1.5 Decision Making

The battlefield scenario is uncertain and changes dynamically. Different models and resolution needs to be consistent with each other. To make the UAV attack efficiently and effectively, proper strategy is needed to direct the UAV in attacking targets. Strategy generation and selection, models changing, and resolutions updating are all based on decision making.

Decision science contributes both to the understanding of decision making and to developing methods and tools to assist that decision making (Davis *et al.*, 2005). Decision making is important to the selection of a course of action among variations. Every decision making process produces a final choice. It can be an action or an opinion. It begins when we need to do something but do not know how to do it.

Rational choice theory, also known as rational action theory, is a framework for understanding and often formally models social and economic behavior. Although models of rational choice are diverse, all models assume that individuals choose the best action according to stable preference functions and constraints facing them. When human makes decision, there are many uncertain and complex conditions, and the decision needs to be made within limited time and resource [4].

The empirical work of Gary Klein (1997) on expert behavior in high-pressure environments resulted in a new school of thought in decision-making. This simulation uses Naturalistic Decision Making theory. The naturalistic decision making (NDM) framework emerged as a means of studying how people actually make decisions and perform cognitively complex functions in demanding situations. The NDM paradigm argues that people assess situations by using prior experience and knowledge. This simulation uses Bayesian Network to support decision making.

1.6 Situation Awareness

Situation Awareness is an important technique for decision making which deals with complexity, complicate, dynamical and uncertain situations. It should perceive the situation before making a decision. If there is no perception or the perception fails, the decision making may not right and may cause the failure of the action. In this simulation, a perception model is implemented in the Filter to help UAV accurately perceive the situation.

1.7 Integration Mulisimulation with Decision Making Support Subsystem

Symbiotic Simulation interacts with the physical system in a mutually beneficial way. The simulation system not only performs “what-if” experiments that control the physical system, but also accepts data and responses to physical system. In this simulation, Multisimulation cooperates with the decision making supporting subsystem. Models updating and switching of stages, resolutions switching of levels and stages are all supported by the decision making subsystem.

1.8 Unmanned Aerial Vehicle

An Unmanned Aerial Vehicle (UAV) (See Figure 1.1) has no onboard pilot. It flies autonomously based on pre-programmed flight plans or more complex dynamic automation systems. Typically they are controlled by a remote operator or autonomous control logic. UAV has several main functions remote sensing, interaction, and transport. Remote sensing includes electromagnetic spectrum sensors, biological sensors, and chemical sensors. This simulation only simulated basic electromagnetic spectrum sensing function for each UAV perceiving the battlefield situation. Interaction includes attacking, taking samples, delivering a payload, and repairing a given structure. This simulation provides the attacking function for UAV and new functions such as interaction between UAVs and the controller, team formation, and team action with strategies. This simulation does implement the UAV transport payload. The UAVs in this simulation work cooperatively, exchange data, and communicate with each other in a timely manner. They can form a team to attack a group of targets based on some strategies. Their actions are evaluated and strategies are updated based on the evaluation.

An Unmanned Aerial Vehicle is widely used in the battlefield and it has many advantages. Single UAVs have been used in the battlefield for years. While in some conditions such as attacking a group of targets, more vehicles need to work together to attack in an efficient and effective manner. In this simulation, UAVs are generated at Matlab simulink. Their initial task is sense or to perceive the area looking for the target. A 2D visualization is generated by Simulink to show this scenario. The data sensed by the UAV is sent to the Filter which is simulated by Visual C++.Net to analyze and integrate to get the full picture of the battlefield. When a group of targets is found, a team of UAVs is formed to attack them. A strategy for the attacking team is generated by Tactical Federate simulated by Visual C++.Net. It selects a suitable strategy to make the team work efficiently and effectively. It launches the high resolution with the team information, strategy, and targets information. High resolution simulates the UAV team attacking a group of targets and shows the scenario in a 3D visualization.

1.9 Matlab Simulink

In this simulation, Matlab Simulink is used to implement the low resolution and generate a 2D visualization.

Matlab is a high-level language and interactive environment that enables the user to perform computationally intensive tasks faster than with traditional programming languages such as C, C++, and FORTRAN. Simulink is integrated with Matlab and provides multiple visualization functions to programming. Simulink is an environment for multidomain simulation and Model-Based Design for dynamic and embedded systems. It provides an interactive graphical environment and a customizable set of block libraries that the user can design, simulate, implement, and test a variety of time-varying systems, including communications, controls, signal processing, video processing, and image processing. Simulink provides an environment where user can make model of physical system and controller as a block diagram. The block diagram is created by using a mouse to connect blocks and a keyboard to edit block parameters (see Figure 1.5). Matlab can generate visualization window for simulation and it provides multiple functions to control the visualization process.

In Simulink, when the UAVs are launched, they unfold the battlefield by sensing it from west to east. They fly on the initialize routes. Simulink exchanges UAV and target information with Filter in a timely manner. Based on the information, Filter clusters the targets as a battalion. When a battalion is found, UAVs form a team. All team members will fly to the center point of the battalion. After attacking, they disperse and attack the rest of the targets. This scenario is synchronized shown in the 2D visualization generated by Matlab.

1.10 Filter

1.10.1 Introduction

Filter module is simulated by MS VC++.Net. It filters the data from Matlab. It communicates with Matlab Simulink and exchange data with it. When the UAVs are launched, every UAV's private information such as position and perception of the field are sent to Filter. Based on the information, Filter integrates data and analyzes data to unfold an entire picture of the battlefield. It analyzes the targets by clustering them based on the *cluster algorithm*. Attacking a battalion is hard and dangerous for a single UAV. In this simulation, if there is a battalion, a team formation process is invoked, and one or more UAV teams are formed. A group of multi-UAVs in close formation looks like a flock of birds (see Figure 1.6) and relies on many technologies such as advanced control methodologies [16]. The team formation process is based on *Contract Net protocol*. It allows contracting as well as subcontracting. Since each UAV in the team could have different perceptions to other team members, a consensus of perception within the team is confirmed after the team formation. Filter sends the team and battalion information to Tactical Federate for further use. The Filter also serves as a communication channel between Simulink and Tactical Federate. It passes UAVs and targets information from Tactical Federate to Simulink making their models consistent.

Protocol used for communication between Simulink and Filter is UDP. Because Matlab integrates C within it and Filter is implemented by C, they can use UDP to communicate implemented by C. Filter and Tactical Federate communicate also based on UDP.

1.10.2 Contract Net protocol

Contract Net protocol is proposed by Smith (1980) to solve the problems in the distributed, multi-agents system such as tasks allocation. It is a fully automated negotiation protocol and work like the electronic marketplace for buying and selling goods. There are two types of agents, Initiator and Participant. At any time, any one agent can be an Initiator, Participant or both. In this simulation, the team leader serves as the Initiator and team members are the Participants.

1.10.3 Cluster Algorithm

Clustering is the classification of objects into different groups, or more precisely, the partitioning of a data set into subsets (clusters), so that the data in each subset (ideally) share some common trait - often proximity according to some defined distance measure. In this simulation, modified K-means algorithm is used. The K-means algorithm is an algorithm to cluster objects based on attributes into k partitions. It is similar to the expectation-maximization algorithm for mixtures of Gaussians in that they both attempt to find the centers of natural clusters in the data. It assumes that the object attributes form a vector space. The objective it tries to achieve is to minimize total intra-cluster variance, or, the squared error function $J = \sum_{j=1}^{k} \sum_{i=1}^{n} \left\| x_i^{(j)} - c_j \right\|^2$, where $\left\| x_i^{(j)} - c_j \right\|^2$ is a chosen distance measure between a data point $x_i^{(j)}$ and the cluster center c_j. J is an indicator of the distance of the n data points from their respective cluster centers. The algorithm is composed of four steps:

1. Place K points into the space represented by the objects that are being clustered. These points represent initial group centroids.

2. Assign each object to the group that has the closest centroid.

3. When all objects have been assigned, recalculate the positions of the K centroids.

4. Repeat Steps 2 and 3 until the centroids no longer move. This produces a separation of the objects into groups from which the metric to be minimized can be calculated [55].

1.11 Tactical Federate

Tactical Federate makes the decision of what action the UAV team will perform by selecting a strategy for the team. It engages the team federates and launches the high resolution. It communicates with Filter and High resolution. It forwards information exchanged between low resolution and high resolution to dynamically update them and to make them consistent. It is better to attack a battalion by UAV team than by a single UAV. If there is no strategy for the team, the UAVs will attack the targets randomly and may not be efficient or effective. A strategy is needed for this scenario. There are two strategies that can be used for a team in this simulation. Tactical Federate will select the proper strategy for the team at specific situations to make its attack efficient and effective. Bayesian Network is used for making decisions to select the strategy. A Bayesian network (or a belief network) is a probabilistic graphical model that represents a set of variables and their probabilistic dependencies. Based on the calculation of Bayesian Network, the probability of using each strategy is given and the strategy with the larger probability is selected as the strategy for team attack. The strategy is evaluated at each period of time during the attack by recalculation of the Bayesian Network. The strategy with the larger probability is selected for the UAV team which is acting at high resolution.

Tactical Federate communicates with high resolution use HLA RTI provided by Mak. HLA RTI is based on UDP. Tactical Federate sends information to high resolution to build a scenario of the UAV team attacking a battalion. The high level resolution also sends back UAV team and battalion information to Tactical Federate for evaluating the current strategy and for updating the 2D visualization which is generated by Matlab. The information is sent back to Filter, and then to Simulink to update the visualization of low resolution. Tactical Federate also sends information to Low Resolution Visualization module (LRV) to generate a 2D visualization.

1.12 High Resolution

1.12.1 Introduction

High resolution simulates the scenario of the UAV teams attacking battalions. It generates a 3D visualization to show this UAV team action. UAVs attack battalions based on the strategy generated at Tactical Federate. Since the battle field scenario is dynamic, and the UAV or target may be destroyed at any time, the strategy may not always be efficient and effective during the attacking. Strategy is needed to be evaluated and updated during the attacking. Since high resolution sends the battlefield information to Tactical Federate, strategy can be evaluated at each period of time based on the information. After evaluating the strategy, Tactical Federate sends back the proper strategy for this period of time. The UAV teams in high resolution will use the strategy sent from Tactical Federate. High resolution is simulated by VC++.Net using HLA Mak RTI and VR-link. It communicates with Tactical Federate using the HLA RTI which is a product provided by Mak. The visualization is generated by Stealth which is a product also provided by Mak. In this simulation, there are multiple high resolutions with their respective visualizations generated by Stealth. Different teams or one team using different strategies are simulated at their respective high resolutions.

1.12.2 VR-Link, HLA Mak RTI, Stealth

VR-Link, HLA Mak RTI, Stealth are products provided by Mak and used for high resolution simulation which simulates UAV team attacking battalion and shows a 3D visualization.

VR-Link

VR-Link is a toolkit. It can quickly and easily use network simulators and virtual reality applications using either the U.S. DoD's High Level Architecture (HLA) or the Distributed Interactive Simulation (DIS) protocol.

HLA Mak RTI (Run Time Infrastructure)

High Level Architecture (HLA) is a generally purpose architecture for distributed computer simulation systems. Using HLA, computer simulations can communicate to other computer simulations regardless of the computing platforms. Communication between simulations is managed by a run-time infrastructure (RTI).

Mak RTI is the fastest way for High Level Architecture (HLA) compliant simulations to communicate. This efficient and easy-to-use tool eases concerns about the performance of the HLA by keeping CPU, bandwidth, and memory requirements to an absolute minimum. It is suitable for battlefield simulation.

Stealth

Stealth is a 3D visualization tool that focuses on information. It provides the most data about user's networked virtual world, and presents it in a clear and easily accessible way.

1.13 Strategy for UAV Team

This simulation makes the UAV team acting effectively and efficiently supported by using proper strategy in specific situations. The simulation strategy should be based on the experts make decision with their experiences that would improve efficiency and effectiveness of the simulation. There are three unique strategies that are used for attacking. One is Fringe Point strategy, one is region strategy and another is Destroy Communication strategy. To determine the effectiveness of alternative Course of Action (COA), the simulation is dynamically updated and branched with simultaneous execution of simulations, potentially at different levels of resolution. Strategies are dynamically evaluated and may be switched at each period of time to make the UAVs act efficiently and effectively through high resolution.

Figure 1.1 Unmanned Aerial Vehicles (UAV). Predator

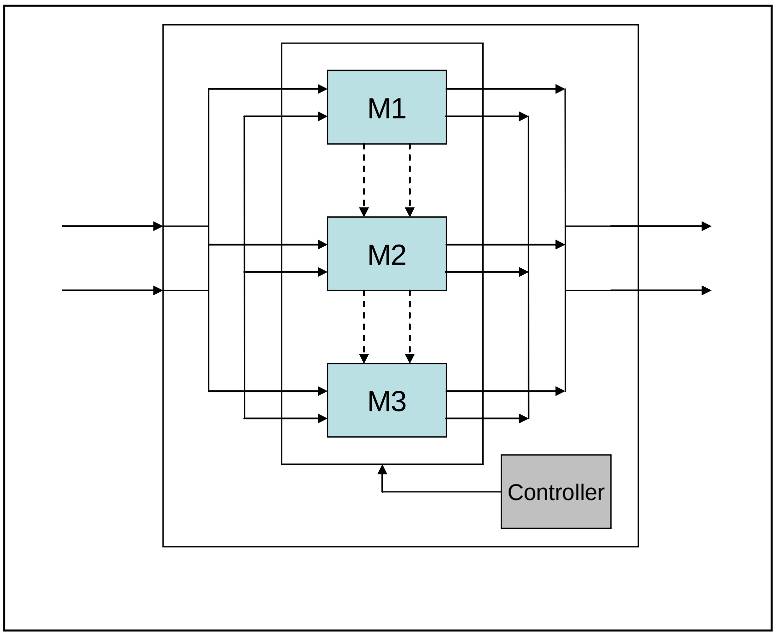

Figure 1.2 Multiresolution Model

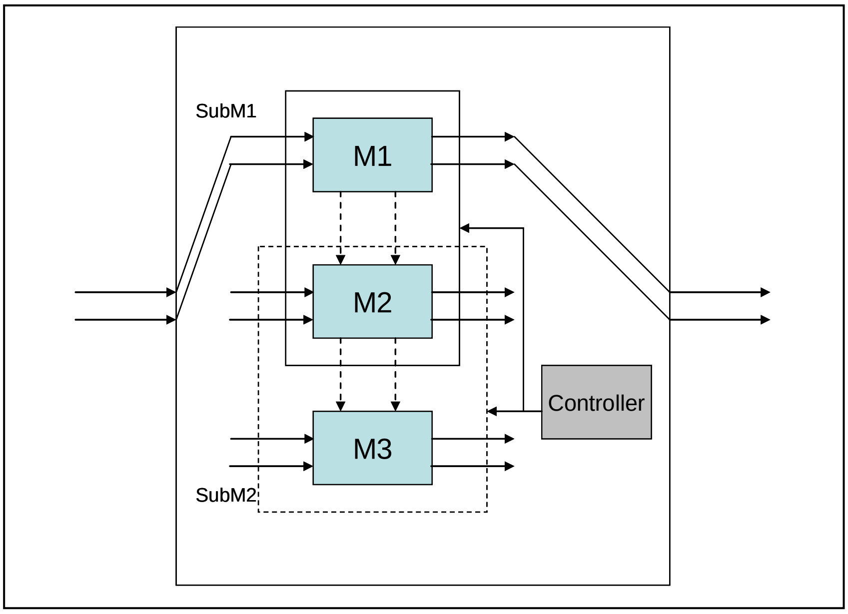

Figure 1.3 Mulitstage Model

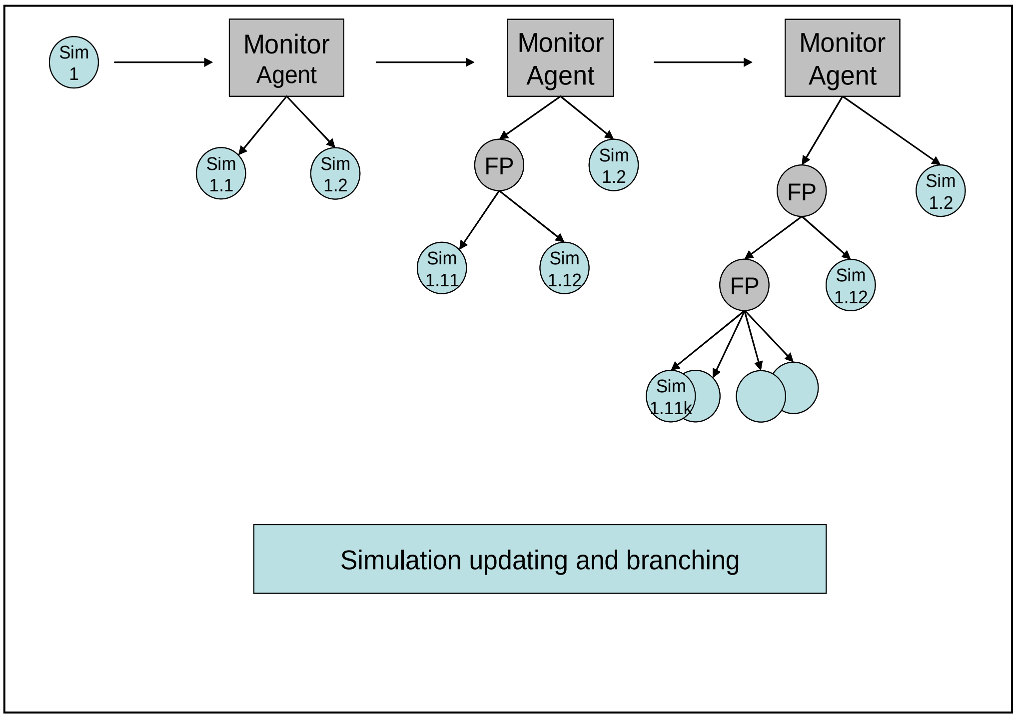

Figure 1.4 Simulation Updating and Branching for Multisimulation

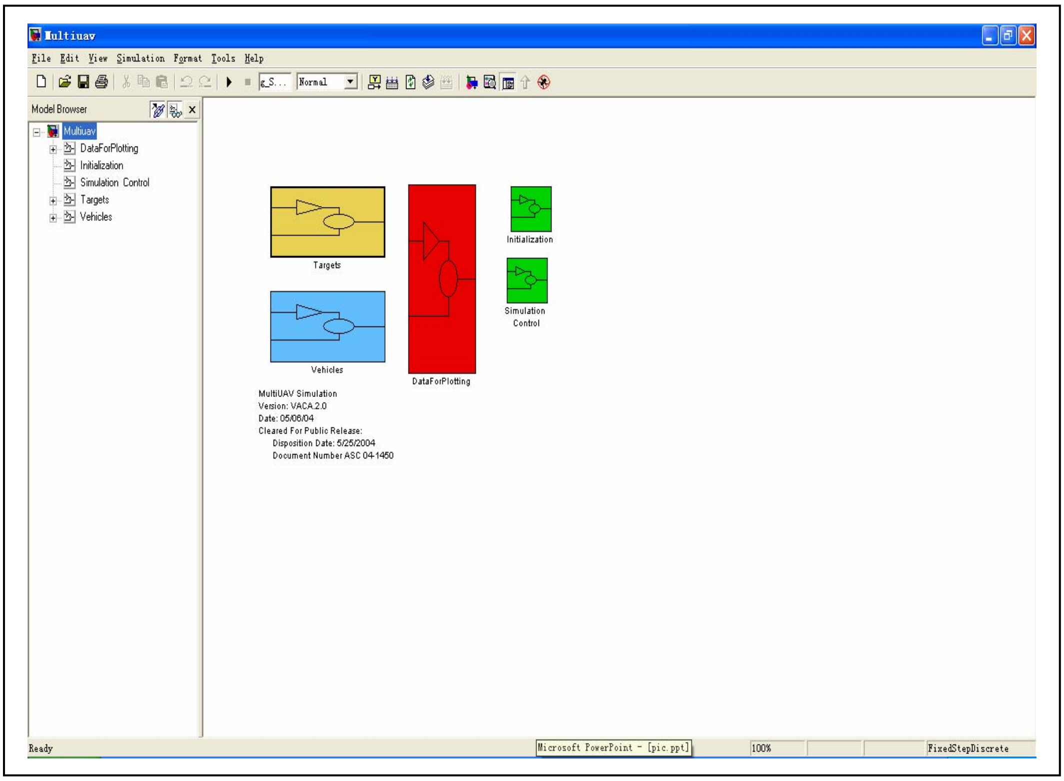

Figure 1.5 Matlab Simulink

Figure 1.6 Formation Flight of UAV Team and Flock of Birds (Song et al., 2005)

CHAPTER 2: MOTIVATIONS

This simulation simulates the multiple UAVs act multiple works including team attacking. These works need to be done efficiently. It is a multisimulation, multresolution multistage system supported by the decision making subsystem and contains several subsystems, resolutions and simulations. These resolutions and subsystems with decision making make the UAVs team work efficiently and different works can be performed at different simulations. The motivation is to make the subsystems run independently on different platforms and work parallel and cooperatively. Another motivation is to integrate the simulations with the decision making system. Each sub-simulation needs to work cooperatively with each other and exchange information in a timely manner. The same model of entity needs to run at different levels of resolutions and consistency should be maintained. The models need to be updated dynamically at runtime, and injected into the simulation, or stopped. These actions are based on the decisions made by the decision making system. These are also motivations of this simulation.

There is a 2D visualization for low resolution and a 3D visualization for high resolution. These visualizations should be consistent with each other. That requires exchange update information between the two resolutions and they are dynamically updated.

The adaptive simulations need other support by the decision making system such as targets clustering and team formation. Targets are needed to be clustered as battalion based on cluster algorithm. A single UAV can not attack a battalion by itself. A team is necessary for the UAV to attack a group of targets. Strategies are used for the UAV team work. Because of the uncertainty of the scenario, strategies are needed to be evaluated at each period of time and updated to select the proper strategy for specific situations. That makes simulation working efficiently and effectively through the simulation. A naturalistic decision making system is needed to make a decision for the UAV action such as strategy selection at different simulation stages and levels of resolutions. These are also the desirable applications and motivations of this simulation.

CHAPTOR 3: OBJECTIVES

This simulation contains multiple simulations, resolutions and subsystems. They are supported by the decision making system. The object of the simulation include making the parties cooperate work with each other, each party can be supported by the decision making system, UAVs can achieves efficiency and effectiveness when they work. The object also includes implements two resolutions with multiple stages, each resolution has its visualization to illustrate the scenarios. We need to implement strategies for UAV team in different situations. These strategies are evaluated based on the simulation results. The result of the evaluation should be that UAV team uses the proper strategy in different situations and always works effectively.

To achieve these objects, some algorithms need to be made such as clustering algorithms to cluster the targets, and consensus algorithm. The simulation scenarios we need to achieved are: the UAVs are launched at low resolution starting searching the battlefield; they sense the area and exchange data with the controller; they cluster the targets they found; when the group of targets is a battalion, they will form a team led by the team leader and make a consensus within the team members; they attack the battalion based on a strategy which is selected by the decision making system, and the strategy will be updated based on the run-time situations. This will make them attack in an efficient and effective manner.

The low resolution is presented in a 2D visualization and high resolution is presented in a 3D visualization. The two visualizations are also consistent with each other. The consistency is based on the timely information exchange between the resolutions.

CHAPTER 4: CHALLENGING PROBLEMS

4.1 The Problem of Deep Uncertainty

This simulation simulates a battlefield scenario. In the most realistic scenario, the situations dynamically changes when the scenario unfolds. Some scenarios may not use any decision style. Dynamic adaptive in decision making is necessary to deal with emergent conditions and make the decision making process flexible. This requires implementing the naturalistic decision making algorithm. Actions should be evaluated at each period of time to select the right strategy. Models should be updated at run-time. Each model needs to be consistent with the others. To fulfill these requirements, the following issues needs to be resolved.

4.2 Multiple Simulations at Multiple Stages

This is a project having multiple simulations, resolutions and models in which different resolutions works at different platforms. Making the models and levels of resolution consistent with each other is a big challenge in this simulation. Low resolution is simulated by Matlab Simulink. Filter and Tactical Federate is simulated by VC.Net. They are simulated by different software by using different programming language. They could be run on different machines or platform, and are required to cooperate with each other and exchange data through the network. There should be a policy to enforce consistency between the parties. This requires a communication channel to connect these parties to share the information. Decision making for UAVs in the battlefield requires perceiving situations accurately. This requires the UAVs to sense the battlefield in a timely manner and work collaboratively with each other. The decision making system should have the function to analyze the data perceived by each UAV to make a decision.

4.3 UAV Team Work with Strategy Support and Targets Clustering

It is dangerous to a single UAV to attack a group of targets. The UAV team is needed in this situation. A team can accomplish tasks that single one can not afford. When a team is formed, the problem is which targets it will attack. Targets need to be clustered as battalions and attacked respectively. If a battalion is found, the UAV team is formed and attack the battalion cooperatively with a strategy. The big challenge is coordination of multiple UAVs forming and working cooperatively in a system. Some issues arise when a team is formed, such as how many team members should be in a team, who is the team leader, and which battalion should this team attack. Each UAV has its own sensing range and can only detect the targets within its sensing range that cause the difference between UAVs perceptions. A consensus of perceptions is needed to be confirmed within each UAV team to attack accurately. To attack efficiently and effectively requires that decision making system choose a right strategy for the team. Naturalistic decision making algorithm needs to be implemented to select the strategy. To make the right decision to select a strategy, the UAVs and targets information should be considered and analyzed. The information should include number of UAVs and targets, area of the battalion and the communication capability of the targets. The selected strategy is needed to be evaluated periodically, and dynamically updated when the current strategy is not the best one. This process should work throughout the simulation to make the overall simulation effective.

CHAPTER 5: RELATED WORKS

5.1 Distributed Task Allocation

Multi-UAV is a popular research field. Multiple tasks need to be allocated to UAVs. The challenging problem is allocation of the tasks to multiple UAVs. Lemaire *et al*., (2004) presented a distributed tasks allocation scheme in Multi-UAV Context. The tasks the UAVs must achieve are essentially “goto” tasks, which are generated either directly by an operator or a central system that analyses all the information acquired by the various UAVs. It requires maintaining a communication link between the UAVs and the central station. The UAV sensing range could affect task allocation. Sujite *et al*., (2005) presented a negotiation scheme that efficiently allocates tasks for multiple UAVs. The paper presents the effect of sensor ranges on the task allocation and compares the results with that of greedy strategy and a variation of the negotiation mechanism. The result shows that the negotiation based task allocation strategy performs far better than greedy strategy. The result also shows that when the sensor radius is low, negotiation with targets information exchanged between neighborhood UAVs performs better than without the targets information exchange.

When the UAVs can not communicate with each other and the UAVs sensors have limited range to detect the targets and neighboring UAVs, and assess target status, the task is difficult to allocate. Sujit *et al*., (2005) presented a team theory for Multi-UAV task allocation.

The team theoretic approach can efficiently allocate UAVs to the targets under this constraint.

Centralized system has some problems such as scalability with the number of UAVs, network delays when UAVs communicate with the central location, and central location of the system is susceptible to malicious attacks. The distributed system can resolve the problems. Dionne *et al*., (2007) presented a new decentralized task allocation algorithm for UAVs. It uses intermittent and asynchronous communications. The algorithm provides a decentralized task consensus (DTC) across the UAVs. The decision to communicate is based on the outputs of the task allocation procedure to reduce the communication overhead. When the task allocation is decentralized such as each UAV allocate its own tasks, the information available is in general different from one UAV to another. To achieve cooperative behavior with decentralized task allocation approach, a decentralized task allocation algorithm resilient to inconsistent information across the team can be selected. This decentralized task allocation algorithm encourages consensus by asynchronous intermittent communications, each UAV communicates when the error between its local and shared information exceeds a given threshold. The communication objective is to achieve a consensus in the outputs of the task allocation rather than in the input information. The algorithm increases efficiency in the communications. It simulates a team of unmanned almost-lighter-than-air-vehicles (ALTAVs) equipped with local sensors and a communication system. The results shows that compared to decentralized information consensus (DIC) algorithm, the DTC algorithm is demonstrated to require less communications to achieve a similar or better cooperative behavior. Dasgupta (2006) presented a distributed automatic target recognition system using multi-agents UAV swarms. It introduces a system COMSTAR-UAV (COoperative Multiagent System for automatic TArget Recognition using Unmanned Aerial Vehicles). The system uses swarming techniques inspired from insect colonies to perform automatic target recognition in a distributed way.

5.2 Path Planning

Path planning is one of challenging problems in the multi-UAV simulation. Ahmadzadeh *et al*., (2006) presented a path planning algorithm for time critical cooperative surveillance. It uses a set of UAVs and they contain body fixed camera. An Integer Programming based strategy is developed. Integer Programming is applied over a receding planning horizon with terminal cost to reduce the computational effort of the planner and to incorporate feedback. It incorporates highly constrained motion and sensor capabilities of the vehicles in the mathematically programming formalism. Kanchanavally *et al*., (2006) presented a control scheme via feedback linearization for three dimensional cooperative path planning of a class of interconnected systems which is UAVs in this simulation. It showed that the feedback linearization technique along with a distance varied repulsive profile allows UAVs to converge to a known target location without colliding with other. It uses Lyapunov stability analysis and the results show the conditions are stable under the system.

5.3 Consistency Maintenance

Multiple UAVs simulation could require multiple resolutions. Different resolutions are work respectively and need to be consistent; different stages in one resolution also need to be consistent. Maintaining the consistency of the resolutions is important to the simulation. Reynolds *et al*., (1997) presented a paper titled "Consistency Maintenance in Multiresolution Simulations". This paper described that inconsistency may occur despite the existence of valid models at each resolution level. Cross-Resolution Modeling (CRM) attempts to build effective Multiresolution simulations. A common solution approach is to dynamically change the resolution of an LRE (or HRE) to match the resolution of other encountered entities. This dynamic change is defined as aggregation (HREs to LRE) or disaggregation (LRE to HREs). The traditional approach to CRM aggregation-disaggregation causes chain disaggregation and puts an unacceptable burden on resources. Multiple Resolution Entities (MREs) is designed in this paper. MREs are the foundation of a design that incorporates maintaining internal consistency. Maintenance of core attributes is as an approach to maintaining internal consistency within an MRE. MREs make possible the isolation of the issues related to maintaining consistency, and they enable possible efficiencies through concepts such as core attributes. This paper describes the means that maintain the consistency of the resolution with dynamic aggregation-disaggregation and resolve the problems incurred by it such as chain disaggregation, network flooding, transition latency, and mapping problems.

Liu *et al*., (2007) presented some key techniques on the parallel implementation of Multiresolution models (MRM). It explains the problems of interaction conflict in MRM which includes consistency maintenance of different resolution models of the same entity, interaction conflict and the problem of morphism interactions. Concurrent interactions of multi-resolution models are considered seriously both in syntactically and semantically. The paper presented two new concepts, named morphism attribute and morphism event. Morphism attributes mean representation of the inherent attributes of an entity at deferent resolutions. Morphism events mean an essentially same event emitted by models of different resolutions, also called morphism interactions. To solve the inconsistent problems, it presented three methods, mutex locking, conflict detecting and model-oriented method. It is a progress in using Multiresolution models in simulation. It enhances the validity of simulation and engagement fairness in the battlefield simulation.

5.4 Multi-agents System

Multi-agent techniques have been used in multi-UAV simulation and improved the simulation. Barber *et al*., (2005) presented a run-time multi-agents system which includes the sophistic agents' interactions, uncertain environment conditions and dynamic domain requirements. It provided some tools for the design and analysis of agent systems and implementations. The design tools help quickly select agent functionality, assign it to model and evaluate the agents' technologies. Trace is an analysis tool and is provided to evaluate the difference between designed behaviors and implemented behaviors. Some technologies are used such as belief revision, information source selection, and action selection. The agents use these technologies to provide a reliable information network and to support agents' action-selection in an UAV target tracking simulation [23]. Karim *et al*., (2005) presented an agent-based autonomous mission control system for unmanned aerial vehicle (UAV). It explores two different approaches for adding autonomous control to an existing UAV. The first adds a layer over the flight control system to control the mission. The second uses human metaphor of agency and implements an autonomous controller based on a model of human decision making. The experiments showed that they are both feasible and useful to a human cognitive modeling approach providing autonomous UAV control [24].

Jang *et al*., (2005) presented two scalable dynamic agent distribution mechanisms for large-scale UAV simulations. One mechanism aims at minimizing agent communication cost, the other mechanism attempts to move agents from overloaded agent platforms to lightly loaded platforms. The mechanisms are fully distributed algorithms and analyze only coarse-grain communication dependencies of agents instead of their fine-grain communication dependencies. In this system, each computer node has one agent platform, which manages scheduling, communication, and other middleware services for agents executing on the computer node.

It simulated large-scale micro UAV simulations involving up to 10,000 agents using the agent distribution mechanisms. The results show that this multi-agent platform is adaptive to improve the scalability of the entire system [25].

5.5 Multi-agents Market-based System

Market based system is widely used for task allocation. It can support allocation of UAVs tasks in the multi-UAV system. Chen *et al*., (2006) presented a market-based approach as the main mechanism for autonomous collaboration in the Autonomous Collaborative Mission Systems (ACMS). ACMS is an extensible architecture and behavioral planning / collaborative approach to achieve the requirements for UAV in Future Combat System (FCS) Unit of Action (UoA). The market-based collaboration approach has a two-stage task specification and negotiation process that can accommodate different mission planning and task allocation strategies. It provides flexible collaboration among a variety of heterogeneous UAVs for a broad range of missions [29]. Dasgupta *et al*., (2006) presented a market based distributed task allocation across swarm units comprising a swarm-based system. The individual swarm units use the task selection mechanism to determine the order of executing tasks in the environment. It uses a computationally simple, yet efficient dynamic pricing algorithm to solve the distributed task selection problem for swarming in the market based model.

The simulation result showed that the market based task selection strategy performs better than other heuristic based task selection strategies and can reduce the communication overhead [30].

5.6 Genetic Algorithm

Since the battlefield scenario is dynamic and evolutional, genetic algorithm is useful to deal with the battlefield situations to support the UAVs acitons. Price *et al*., (2006) presented genetic algorithm directed self-organized search and attack UAV swarms. The paper investigated the use of a self-organization (SO) framework for evolving UAV swarm behavior. A genetic algorithm (GA) is used for UAVs evolving behavior to successfully locate and destroy stationary targets. To make the Multi-UAV systems autonomous, it uses the organizational concepts used by colonial insects, wolf packs and economics. The self-organization realizes the groups of UAVs as a singular macro system rather than a set of interacting individuals. It provided a SO model for engagement between targets and UAVs, and it demonstrates an architecture for future modeling and SO UAV swarm behavior evolution. The SO model allows for the evolution of a multi-agent system. In most cases, it provides a cooperative and useful set of UAV behaviors [26]. Barlow and Oh (2006) presented an analysis for genetic programming controllers for unmanned aerial vehicles. It assures that prior to operation that an evolved controller will not damage the vehicle. A poorly performing controller might damage the vehicle. Evolved controllers must be robust to noise in the environment to operate the vehicle safely. The navigation controllers of UAVs use multi-objective genetic programming. The simulation results show that best evolved controller performs better than two hand-designed controllers and is robust to many sources of noise [27]. Lamont *et al*., (2007) presented a comprehensive mission planning system for swarms of autonomous aerial vehicles (UAV) using multi-objective evolutionary algorithm.

The system deals with the problems including path planning, vehicle routing, and hierarchical architecture-based swarms' behavior. The system has a path planner which is multi-objective evolutionary algorithm-based terrain-following and parallel, and an evolutionary algorithm-based vehicle router. It aims to minimize cost and risk generally associated with a three dimensional vehicle routing problem (VRP). The system finally is an extensible developmental path planning model integrated with swarm behavior and tested with a parallel UAV simulation [28].

5.7 Evidential Map-building

Yang *et al*., (2005) presented two evidential map-building approaches for Multi-UAV cooperative search. One is based on Bayesian theory and the other is based on Dempster-Shafer theory. It utilizes the generated maps into the UAVs path planning procedure to cooperatively localize targets in the environment. The paper described the problem that a team of Uninhabited Air Vehicles (UAVs) seek to find targets in an unknown and uncertain environment. The decision on where UAVs search next is driven by the aims to increase the chance of locating targets and possibly avoiding threats. For this reason, UAVs need a good model of the environment and they should be able to construct and consistently update their environment models based on the sensory information. The efficiency and quality of the environment model consequently affect the UAVs search planning and decision-making activities. Since the UAVs sensor information of the real world may be inaccurate and uncertain, different vehicles need to combine the sensory information to obtain the best knowledge of the environment. The paper use evidential reasoning techniques to fuse the sensors' information which is used as evidence. It shares the information to each other. The two evidential methods incrementally build cognitive maps to direct a group of UAVs to perform cooperative search tasks. Both of the methods are suitable for sensor measurements obtained in a sequential manner. They also allow the information to be incorporated with formula of finite horizon optimal control for the multi-UAV cooperative path planning problem in the search tasks. The simulation results show that these map building methods are effective to deal with the problem of UAVs cooperative search [31].

5.8 Team Work

In Multi-UAV system, multiple UAVs cooperatively work to fulfill various tasks in the battlefield. Team work is a key point for UAVs to work effectively. Jeyaraman *et al.*, (2005) presented using formal methods to model and verify cooperative unmanned aerial vehicle (UAV) teams. Since the multi-UAV coordination for cooperative control is a time-critical, zero-fault tolerant activity involving dynamic planning and real-time decision making, the formal methods can be used in this application area. It simulated multiple UAVs coordinated arriving at a specified target using Dubins' curves. The simulation is modeled using the Kripke models of "possible worlds". This formal model then used model checking technique to verify safety and reach ability. This model can be reused and be able to address some scalability issues [33].

Gowtham and Kumar (2005) presented a simulation of multi-UAV flight formation. UAVs perform tasks such as surveillance, reconnaissance and target attack etc. Some algorithms have been developed for formation flying. It took into account the static and dynamic obstacles such as wind gusts that are encountered during the mission. It simulates arbitrary numbers of vehicles in a virtual reality environment. The simulation results show that the predetermined path is maintained by alleviating the deviations. In this way, it avoids static and dynamic obstacle collisions, inter vehicle collisions, and maintains the flight of formation [34]. Corner *et al*., (2004) presented a parallel simulation of UAV swarms. The framework includes support for discrete event simulations. This allows for quickly integrating higher fidelity or specialized support models as needed. Speedup is not achieved with the current implementation [15].

Song *et al*., (2005) presented an orthogonal transformation-based robust adaptive control method for close formation flight of multi-UAVs. The method deals with general case of uncertain dynamics due to vortex and external disturbances when UAVs are in close formation. It developed robust and adaptive control algorithms to maintain the desirable separation of wingman with the leading UAV or team leader. Compared with other existing methods, the formation control scheme is simpler in structure and easier to implement. The control scheme does not need detailed system dynamic information [16].

McLain *et al*., (2001) presented the cooperative control of UAV rendezvous. The paper addresses the problem cooperative rendezvous in which multiple UAVs are to arrive at their targets simultaneously. It describes the development of a rendezvous manager state machine and a cooperative control decomposition approach [21].

Goktogan *et al*., (2003) presented the system architecture of a real time Multi-UAV simulator. The simulator is a real demonstration of multiple UAVs conducting both decentralized data fusion and control. It has testing and validation mechanism. The mechanism includes off-line simulation of complex scenarios, hardware-in-the-loop tests, validation of real test results, and on-line mission control system demonstrations. It presented CommLihX, a communication framework. The system allows simulation modules to communicate over single or multiple virtual channels [14].

CHAPTER 6: GENERAL APPROACHES

6.1 Introduction

In this chapter, general simulation scenario and general approaches are described. The models described in this chapter include the low resolution model which is implemented in Matlab, Filter, Tactical Federate and high resolution model. Their general functions and cooperative works are illustrated. Some of the algorithms implemented in this simulation are also introduced.

6.2 Simulation Scenario

This simulation simulates multiple UAVs at the battlefield scenario. The UAV has multiple tasks including search, tracking, forming, attacking, and evaluation. The tasks are allocated based on the rum-time analysis of the UAVs and targets information.

The simulation begins at low resolution which is simulated by Matlab Simulink. There is a 2D visualization for low resolution generated by Matlab. When the UAV is launched, it starts searching the battlefield. Each UAV has its own view of the battlefield based on its sensing.

The UAVs and targets information are sent to the Filter for analysis. Based on a cluster algorithm presented in this article, targets are clustered. The Filter will decide whether the target is single or belongs to a battalion. If a battalion is detected, the UAVs need to form teams to attack the battalion. Based on the Contract Net protocol, a team leader and several team members are selected to do the task. One team is assigned to attack a particular battalion. Because of the different views of each UAV and mistakes may be made by the UAV when it perceives targets, the accuracy of those views need to be assured. This means that the team members within one team need to make a consensus of their views. When a consensus is confirmed within one team, each member gets the consensus model.

The team member will fly to the battalion location to attack. To make the attack in an efficient and effective manner, different strategies are needed to be used in different conditions. The strategies are generated by Tactical Federate. Three basic strategies are designed; one is Fringe Point Strategy, one is Region Strategy and the other is Destroy Communication Strategy. The strategy selection for the team at particular condition is based on the analysis by Bayesian Network. After the team formation and strategy selection, teams are engaged and high resolution is launched which is implemented by Mak VR-Link, HLA, Mak RTI. Team information, strategy information and battalion information will be sent to the high resolution simulation. Based on the information, high resolution simulation is initialized for the multi-UAV teams attack battalions. A 3D visualization is implemented by Mak Stealth to show this scenario.

The team performance at high resolution is evaluated periodically based on the team and battalion run-time information. The evaluation work is done at Tactical Federate. The suitable strategy is selected, and high resolution is updated with it. The low resolution and high resolution need to be consistent with each other such as the active UAVs and targets information in both resolutions. The low resolution will update its models when it gets the information from high resolution. The two resolutions exchange information periodically. To be consistent, models exchange data with each others and are updated at run-time. When the UAVs in the high resolution destroy all the targets or they are all destroyed by the targets, high resolution stops. When the low resolution is informed that high resolution stops, it allocates new tasks to the rest of UAVs. If the battalion is destroyed, the UAV will disperse and go on search the area to find and attack other targets. If the UAV team is destroyed, the rest of UAVs will form new teams to attack the battalion.

6.3 Low Resolution

Low resolution models are simulated by Matlab Simulink. The simulation begins from low resolution. The UAVs and targets are generated and initialized in Simulink. This part simulates the UAV independent action. Different UAVs have different colors. Since the UAV has a limited sensing range, it can’t sense the target outside its sensing range. All targets are initialized having distances between them and UAVs larger than the UAV sensing rang that means all targets are outside all the UAVs sensing range. UAVs can’t see any target at first. The first task of the UAVs is to fly toward the east and sense the area to look for the targets. When the UAVs fly and sense, the area is unfolded, and targets can be found by the UAVs. The targets found by UAVs will be marked black. Low resolution sends all UAVs and targets information to Filter. Based on the feedback from Filter, the targets are clustered and different clusters are marked by different colors in the low resolution visualization. Filter also sends back the new tasks allocated to the UAVs to low resolution based on the analysis of the UAVs and targets information. The UAVs in low resolution then performs the tasks assigned to them.

When a team is required, Filter sends back the team members and task information to low resolution. Based on the information, the UAV teams are formed and fly as a swarm to attack specific battalion which is assigned by Filter. In low resolution visualization, the UAV team members will fly to the battalion center and stop. At this point, high resolution is launched to perform the attacking scenario. The high resolution periodically sends back UAVs and targets information to low resolution and it dynamically updates the corresponding models.

When the high resolution stops, low resolution is informed. It assigns new tasks to the active UAVs based on the information from high resolution. If the battalion is destroyed, the UAVs will go on searching the rest of the area to find the targets and attack. If there are active targets in the battalion and the UAV teams are destroyed, new teams may be formed to attack the battalion. A 2D visualization is generated by Matlab to show the scenarios of low resolution (see Figure 5.1).

6.4 Filter

The function of the Filter includes: exchange information with low resolution and Tactical Federate, clustering the targets, making consensus model, and form UAV teams. The data received by Filter from the low resolution includes the UAVs and found targets information. The UAV information includes the UAV position and status. The target information is the position of the target which is found or seen by a UAV. Based on the target information, the analysis is processed. The targets are clustered using the cluster algorithm. The cluster function is a function by the distance between the targets. Each cluster is deemed as a battalion. Because of the different perception of each UAV, a consensus model is used to help assure the accuracy of the perception. Each team member gets the consensus model after confirming the consensus within the team. Filter sends back information such as UAV teams, target clusters and consensus to low resolution to update it. Filter also sends this information to Tactical Federate to select a strategy for each team. Filter also received information such as UAV and target from high resolution through Tactical Federate and sends the received information to low resolution to update it. In this way, the two resolutions can be consistent with each other.

6.5 Tactical Federate

The Tactical Federate functions as a control and conjunction between several models. It makes the decision of which strategy to be used by a particular UAV team to attack a particular battalion. This decision making model makes sure the UAVs attack in an efficient and effective manner. This decision making model is based on Bayesian Network algorithm. The UAV team, targets and the battlefield information are the input of the decision making model, and the output are the different strategies with their respective probabilities of being used. This probability shows the chances of using the strategy in the particular conditions. Larger probability means having more chances to get the task done by using the strategy in the current situation. The strategy with the larger probability is selected for the team.

Tactical Federate engage the teams and launches the high resolution model after it selected the strategy for teams. It initializes the high resolution with teams, strategy and battalions information. During the high resolution simulation, teams and targets information are sent back to Tactical Federate as evidences to evaluate the strategies. Tactical Federate evaluates the strategy by using the information as the input of the decision making model, and gets the probability of each strategy at the current situation. Based on the evaluation, strategy with larger probability is selected and high resolution updates the UAV team action based on the selected strategy.

The Tactical Federate also serves as part of a communication tunnel between low resolution and high resolution. In a timely manner, it sends all the UAVs and targets information to Filter, and then Filter sends it to low resolution. When low resolution gets the information, it updates the corresponding models at run-time. In this way, low resolution and high resolution are consistent with each other. Tactical Federate also sends all of the UAVs and targets information to a 2D visualization for visualizing the low resolution.

This visualization is initialized by the information from low resolution and Filter, and then updated by the information from high resolution when the high resolution executes.

6.6 High Resolutions

High resolution is the simulation for the UAV team action with strategy. When it is launched, it is initialized with the UAV team, strategy and battalions information. In this part of simulation, the UAVs fly in formation and attack the assigned battalion with a strategy. UAV team members fly directly to the battalion, and attack the battalion based on the selected strategy. High resolution has multiple simulations and simulates multiple UAV teams attacking different battalions with particular strategies. All UAVs and targets information is sent to Tactical Federate periodically to evaluate the current strategy. After the evaluation, a new strategy may be selected for the team or keep the current strategy. The communication between Tactical Federate and high resolution is implemented by HLA Mak RTI and VR-Link which provide run-time and timely communication between the two parties. A 3D visualization for high resolution is implemented by Mak Stealth. It provides a dynamic 3D animation environment with high quality pictures (see Figure 5.2).

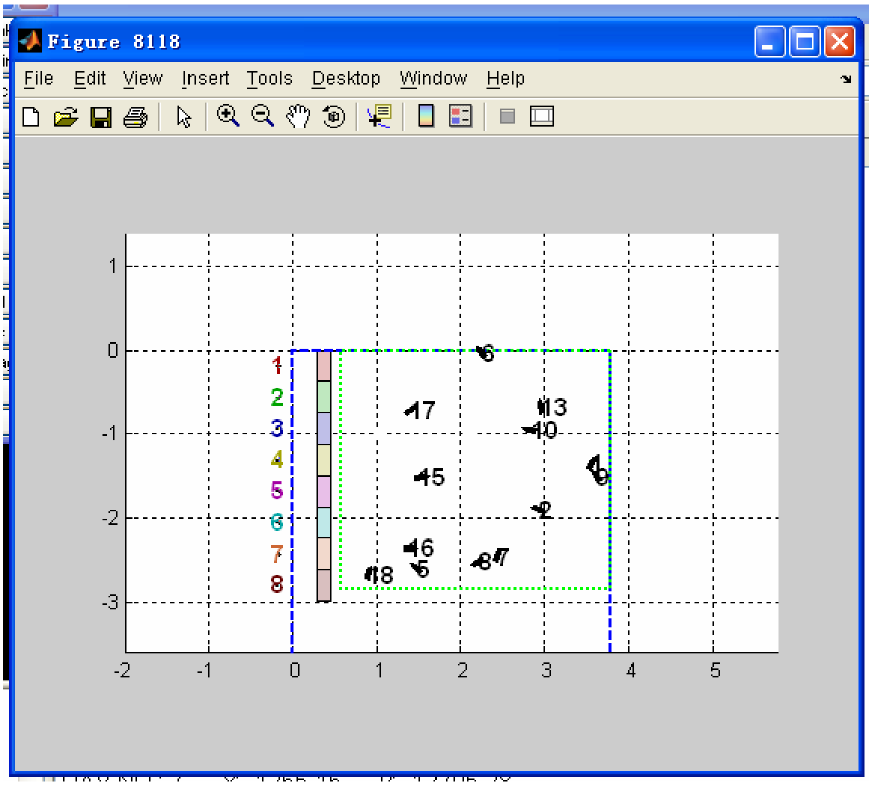

Figure 6.1 Low Resolution Visualization

Figure 6.2 High Resolution Visualization

CHAPTER 7: DETAILS OF DESIGN AND SIMULATION

7.1 Introduction

This chapter describes the design and simulation details. Low resolution, Filter, and Tactical Federate are illustrated with design architecture and implement details. Techniques utilized in those parts are also illustrated in detail.

In low resolution, the UVAs are initialized and looking for target. Every time unit, the sensing information need to be sent to the Filter to analyze. The transmission protocol such as TCP and UDP can be used. Because the communication is run-time and time sensitive, UDP is chosen in this project. Voiding overhead of checking of whether every package arrived, UDP is faster and more efficient.

In the Filter, targets information needs to be analyzed. The targets need to cluster. There are several clustering algorithms we can select. K-means is one of the simplest unsupervised learning algorithms that solve the well known clustering problem [10]. Because the K-means algorithm is sensitive to the initial randomly selected cluster centers, we use modified K-means algorithm to initialize each target as a cluster center. The team formation process is done at Filter. Because each UAV is an individual unit, it can choose whether or not to join in a team. Team leader need to ask the UAV to join in its team.

Contract Net protocol is a good protocol for this scenario since it works like electronic market buying and selling goods. UAV can decide whether to join the team by itself based on this protocol.

Tactical Federate is the strategy decision generator. The strategy is used for the UAV team which attacks a battalion. Because the battlefield scenario is dynamic that makes the evidences such as number of target variable. Some of the evidences are not independent and affect each other. The different strategies may not absolutely good or bad for a specific scenario. We need a model to analyze the evidences of the battlefield and select a proper strategy for the team. Bayesian Network can be used to learn casual relationships and hence can be used to understand a problem domain and predict the consequences of intervention [34]. It is a directional probability dependence graph, using the evidences to analyze and get the probabilities of the causes to the evidences. Because the model has both a causal and probabilities semantic, it can combine the prior knowledge and data [34]. Some of the evidences can not be used for a specific scenario such as efficiency of the team since at the beginning of the attraction there is no efficiency. Because the model of Bayesian Network encodes dependencies among all variables, it readily handles situations where some data entries are missing [34]. Bayesian Network supports this situation and can analyze using all or some of the evidences.

7.2 Low Resolution

Low resolution is simulated by Simulink. Two main models, initialization model and communication model are described.

7.2.1 Initialization Model

When the simulation starts, the initialization model is launched. In this model, the simulation environment is initialized including the battlefield size, the number of UVAs and each UAV's information, the number of target and each target's information. UAV information includes x and y position, speed, sensing range, waypoint, and color. Way point is generated by the way point model which is the functionality model for generating UAV way point. It calculates way point based on the UAV current location and destination location. Target information includes x and y position. In this simulation, there are 8 UAVs and 20 targets. All UAVs start from west of the battlefield and fly at the same speed (see Figure 7.1). The initialized way point of each UAV goes straight from the starting point to the east of the battlefield. Each UAV has its unique color. The targets are set all over the battlefield. All of them are black. The targets are unseen or transparent when the distance between them and each UAV is larger than the UAV sensing range. Since all the targets are set to be out of the sensing range of all UAVs at initialization, all targets are unseen to the UAVs and unseen in the low resolution visualization. When UAVs unfolds the battlefield and the distance between the target and an UAV is less than the UAV's sensing range, the target is sensed by the UAV and appears in the low resolution visualization marked black (see Figure 6.1).

7.2.2 Communication Model

The communication model is one of the most challenging models in this simulation. Low Resolution runs in Matlab Simulink environment, which uses the Matlab language. On the other side, the filter runs in Visual stdio.net environment written in C. Since UDP is the protocol of communication, it is needed to be implemented in Matlab environment. Although there is no UDP program written in Matlab language to communicate to UDP written in C, Matlab has the friendly functionality for using C or C++ within its environment. The code written by C or C++ can be compiled in Matlab. In Matlab, the S-Function, User-definable block can be written in C program and be used after the C program is compiled. In this way, the UDP code written in C can be used in Matlab. The communication model is built in the "Save Plot Data" block (see Figure 7.2). The transmitting data block is "trans" and the receiving block is "recv" (see Figure 7.3). They are S-Functions containing a function *mdlUpdate* which will be executed in each time unit. The time unit in this project is set to 0.1 second (see Figure 7.4). The UDP piece of program is added to the *mdlUpdate* function and is executed in each time unit to send and receive data message between Simulink and Filter. In the Filter, there is a corresponding piece of UDP program to communicate with the UDP program in Simulink. In this way, Matlab program can communicate to the program written in Visual stdio.Net by using the UDP program written in C. The program sequence chart of communication between Simulink and Filter is shown in Figure 7.5. The program execution snapshot is shown in Figure 7.6.

7.3 Filter

7.3.1 Introduction

Filter analyzes the targets, clusters targets as battalion, forms UAV teams, and makes consensus within each UAV team. It also communicates with low resolution and Tactical Federate to exchange information in each time unit.

7.3.2 Targets Clustering

Since the targets are all out of the each UAV's vision range, there is no target sensed by UAV. During the unfolding of the battlefield, the UAVs fly across the battlefield and the target is sensed when the distance between it and the specific UAV is less than the UAV sensing range. Targets data such as x and y positions are sensed by the UAVs. When Filter gets the targets information sensed by low resolution, it clusters the targets by using the modified K-means algorithm.

The clustering process starts by assigning each target to a cluster to create 20 clusters if there are 20 targets sensed. The distance between the clusters is the same as the distance between the targets. The process finds the closest pair of clusters and merges them into a new single cluster. There are 19 clusters at this point. It then computes the distance between the new cluster and other clusters to find the shortest distance and merge them to create a new cluster. It then repeats the previous processes until the cluster size reaches the maximum cluster size which is 4. It can also stop when all distances between the clusters are larger than a threshold which means all the clusters are far way from each other and do not need to be merged. The computation of the distance between clusters uses the Average Linkage to calculate the mean distance between two clusters members. The formula of the calculation is:

$$D(r, s) = \frac{1}{n_r n_s} \sum_{i=1}^{n_r} \sum_{j=1}^{n_s} dist(x_{ri}, x_{sj})$$

A simple clustering process is shown in Figure 7.7. When the clustering process stops, each cluster is deemed as a battalion. Different clusters are marked by different colors (see Figure 7.8).

7.3.3 Team Formation

When the targets have been clustered, there are several battalions on the battlefield. A single UAV is not suitable to attack a battalion and UAV team is needed to attack in an effective manner. The first UAV who finds the battalion will be the team leader. The team leader will decide the team size N based on the battalion size. The team leader will send CFP (Call for Proposals) information to N UAVs which have the minimal distance between them and the battalion center. Each UAV has a willing attribute which is a random integer between 1 and 100 showing the willingness to be in a team. The UAV which received the CFP checks the willing attribute to decide to be a team member. If the willing of the UAV is between 0 and 70, it is willing to be a team member. That means there are 70% chances that a UAV can accept the CFP request. If the UAV is willing to be the team member, it accepts the proposal.

It sends acceptance information to the team leader and makes a team contract with team leader. If the UAV is not willing to be in a team, it sends rejection information to the team leader. The team leader informs the team member by sending a Confirm message. This process is shown in Figure 7.9. If there are not enough team members on the team after receiving all feedbacks from the N UAVs, the team leader will send CFP to the other UAVs based on the distance between them and the cluster center. The process stops when there are N UAVs on the team, or the team leader has received all rest of UAVs feedbacks.

The program execution snapshot is shown in Figure 7.10. The number of UAVs on a team can be three, four, or five since there are eight UAVs total. When the first team is formed, the rest of the UAVs form another team based on the same protocol. For example (see Figure 7.10), there are two teams, team 1 and team 2. Team 1 has three team members which are UAVs having ids 5, 4, 6. Team 2 has five team members which are UAVs having ids 1, 2, 3, 7, 8. Team leader in Team 1 is UAV 5; team leader in team 2 is UAV 1.

7.3.4 Consensus Model

Since there are different perceptions of each UAV within a team, a consensus is needed to be made within a team. After a team is formed, the consensus process is loaded. After this process, consensus is confirmed within the team and each team gets a consensus id. This id indicates that the consensus perception is perception of the UAV team member which has the UAV id equaling to consensus id.

The team will attack the targets seen by that UAV. For example (see Figure 7.10), team 1 will attack the targets seen by UAV 5, team 2 will attack the targets seen by UAV 3.

7.3.5 Communication Functions of Filter

The Filter is a middle part between low resolution and Tactical Federate. It communicates to other parts based on UDP. Filter communicates to low resolution is described in the section 7.2. Both Filter and Tactical Federate are simulated by Visual stdio.Net. Filter is written in C and Tactical Federate is written in C++. The communication between them utilizes UDP socket program. The data transferred between them are in the structure *DataTransfer* which includes UAV, Target, Cluster, and Team information. This data is sent to Tactical Federate and high resolution. The program sequence chart of communication between Filter and Tactical Federate is shown in Figure 7.11 and the program execution snapshot is shown in Figure 7.12.

***DataTransfer* structure**

```
struct datatransfer
{
    team uteam[2]; // all teams information
    TARGET ut[TN]; // all targets information
    utrans u[UN]; //all UAVs information
    ctrnas c[TN]; // all clusters information
```

```
    int noc; // number of cluster
};
```

7.4 Tactical Federate

Tactical Federate has the following functionalities: strategy selection for UAV team, strategy evaluation, team engagement, launching high resolution, and communication to Filter, to high resolution, and to low resolution visualization. Strategy selection, strategy evaluation and communications to other parties work at run-time. Communication with other parties is required to exchange information and ensure consistency between the parties. The Bayesian Network is used to select the proper strategy for the UAV team in a specific scenario. The UAV team is engaged in Tactical Federate based on the team information generated by team formation process in Filter.

7.4.1 Strategy Selection

To make the UAV team attack battalion in an efficient and effective manner, a strategy is used to direct the UAV team attacking.

When the UAV team is formed by Filter and it informs the Tactical Federate, Tactical Federate inputs the information as evidences to the strategy decision model to select a strategy for the team. When the UAV team is acting at high resolution, the feedback information from high resolution is used as inputting evidences to evaluate the current strategy, and a proper strategy is selected for the UAV team at current conditions. Bayesian Network is used as the algorithm to select the strategy. We use the Microsoft Bayesian Network tools MSBNx to implement the algorithm in this simulation. MSBNx is a tool for doing that kind of cost-benefit reasoning for diagnosis and troubleshooting. Based on that, it can select an efficient and effective strategy for the UAV team.

The MSBNx implemented in this project is shown in Figure 7.13. The top nodes are the input nodes as the evidences of this model. They are Area, Betweenness, Degree, CommRang (Communication Range), TargetNumber (number of target), UAVNumber (number of UAV), FSEfficiency (Fringe Point Strategy Efficiency), FSEffectiveness (Fringe Point Strategy Effectiveness), DCSEfficiency (Destroy Communication Range Strategy Efficiency), DCSEffectiveness (Destroy Communication Range Strategy Effectiveness). The bottom nodes are DCS (Destroy Communication Strategy) and FS (Fringe Point Strategy) which are the strategies we can choose for UAV team action (see Figure 7.13). Area is the battalion's approximate area. Betweenness is the number of geodesic paths that pass through a node. It is the number of "times" that any node needs a given node to reach any node by the shortest path. Degree is the number of ties to other nodes, row or column sums of adjacency matrix. Communication range is the ability of one node can communicate to other nodes. Since the efficiency and effectiveness are not generated until the UAV team is attacking battalion, the FSEfficiency, FSEffectiveness, DCSEfficiency and DCSEffectiveness have no value before launching high resolution.

The fringe point strategy is that we suppose there is a fringe around targets area and set several points on it. When UAV team attack these targets in the area, each team member will fly to a point on the fringe. Then, the UAV will attack target within its sensing range in the area. After attacking the target within UAV's sensing range, it will attack other targets in this area. The region strategy is that UAV team leader divide the area into several regions. It will allocate each UAV team member a region. Each UAV will attack targets in the allocated region. The basic idea of destroy communication strategy is to destroy the communication path between targets. The path means the communication route between two targets. For a given target, if the degree is low and betweenness is high that means the target is a critical target for other targets to communicate, since there are less paths income and outcome to the target and this target is on many of the shortest paths between other targets (see Figure 7.14). In this situation, destroy communication strategy has a larger probability to be selected. If there is not a critical target in the situation, the fringe point strategy is more preferred since the UAV team attack based on the fringe point strategy has more effectiveness than ad-hoc attack in general situations.

The Bayesian Network in this simulation is built by MSBNx as the following process. First, a diagram is made shown in Figure 7.13. It illustrates the relations of the nodes. If node A decides or is the input to node B, A is parent node of B and there is an arrow from A to B. The number of targets (TargetNumber) and Area nodes decide the spread of the targets (TargetSpread) in the area which means: "large" spread, a small number of targets are dispersed in a large area; "small" spread, a large number of targets are dispersed in a small area. The spread of targets (TargetSpread), the number of the UAV (UAVNumber) and Degree nodes decide TargetsOverPowered node. DCSEfficiency (Destroy Communication Range Strategy Efficiency) and DCSEffectiveness (Destroy Communication Range Strategy

Effectiveness) nodes decide DCSOverall node. FSEfficiency (Fringe Point Strategy Efficiency) and FSEffectiveness (Fringe Point Strategy Effectiveness) nodes decide FSOverall node. TargetsOverPowered, Betweenness and DCSOverall nodes decide DCS (Destroy Communication Strategy) node. TargetsOverPowered, Betweenness and FSOverall nodes decide FS (Fringe Point Strategy) node.

Second, we set uncertainties, the probabilities of various situations of each node in the Assessment module. For example, we set the "number of targets" node to four levels in its assessment table: low, medium, high, critical with the probability as low 0.15, medium 0.35, high 0.35 and critical 0.35 (see Figure 7.15). The set of uncertainty is done at each node in the system. The node having more inputs has the larger Assessment table since the more situations are introduced. For example, the bottom node DCS has three parent nodes: DCSOverall, TargetOverPowered and Betweenness. Because DCSOverall has 2 situations, TargetOverPowered has 5 situations and Betweenness has 4 situations, the DCS has 40 (2*5*4=40) situations total in its assessment table (see Figure 7.16). The red bar means the probability to select this strategy in the situations which are based on the parent nodes conditions, and the yellow bar means the probability not to select this strategy in the situations. The sum of the red probability and yellow probability is 1 in each situation.

The program loads this diagram at run-time, inputting required information and calculating the probability for each strategy based on the diagram. The calculation result is shown in Figure 7.17. The UAV team uses the strategy with the larger probability to attack the battalion. When the UAV team is attacking the battalion, the UAV team and battalion information are sent back to Tactical Federate to evaluate the current strategy in each period of time. The efficiency and effectiveness for each strategy are generated at high resolution and sent back with the other information. The information is input to the decision making model using the Bayesian Network to calculate the probability of each strategy in the current situation. After each calculation, if the strategy with larger probability is different from the strategy currently used, the UAV team is informed to use the strategy with larger probability. If the strategy currently used has the larger probability, the UAV team still uses the current strategy to attack in the next period of time. The evaluation works at run-time and the strategy update is dynamical at high resolution. In this way, the UAV team can attack in an efficient and effective manner during the high resolution simulation. Eff = NoTar/T is used to calculate the efficiency (Eff). In this formula, NoTar is the number of targets destroyed by the team and T is the time the team spends to do the job.

Results show that UAV team using different strategies based on the evaluation having almost same efficiency. That means the team can attack efficiently at different situations.

In simulation, we implemented fringe point strategy with region strategy for UAV team. In the simulation with fringe point strategy, there is a virtual fringe of the targets area and there are four points on this fringe (see Figure 7.18). Each UAV fly to a point of the fringe at the beginning and attack the target close to the point. The UAVs firstly attack the targets close the fringe, and then attack targets in the inner of the fringe (see Figure 7.19 and Figure 7.20).

Region strategy is based on the dividing the targets into several regions. Each UAV will deal with different regions in this simulation. A region is created by using a single point as a reference. Around the reference, the region is created. The algorithm to create region is:

1. Pick a reference point;

2. Get next N amount of closest targets from the reference point.

3. Aggregate these targets as a region.

4. Repeat from step 1 to 3 until all targets have been put into a region.

Based on the region strategy, the targets are divided into regions and each UAV is assigned to attack a region. When a UAV destroy all targets in its region, it will most likely move to the next closest region. In this simulation, we set N is 2 or 3 which means there are two or three targets in one region. In figure 7.21, there are nine targets and they are divided into four regions circled by different colors. There are four UAVs to attack these targets and each UAV is assigned to attack the targets in a region. When a UAV finishes its work in the region, if there are targets left in other regions, it moves to the closest region among those regions.

7.4.2 Communication Functions of Tactical Federate

Tactical Federate communicates to Filter, Team Federate, high resolution and low resolution visualization. The program sequence chart is shown in Figure 7.22. The communication between Tactical Federate and Filter, Tactical Federate and low resolution visualization are based on UDP. The communication between Tactical Federate and high resolution uses the VR-Link and HLA Mak RTI provided by Mak. All parties communicate and exchange data with each other in a timely manner during the simulation. This periodically data exchange ensures that the parties are consistent with each other.

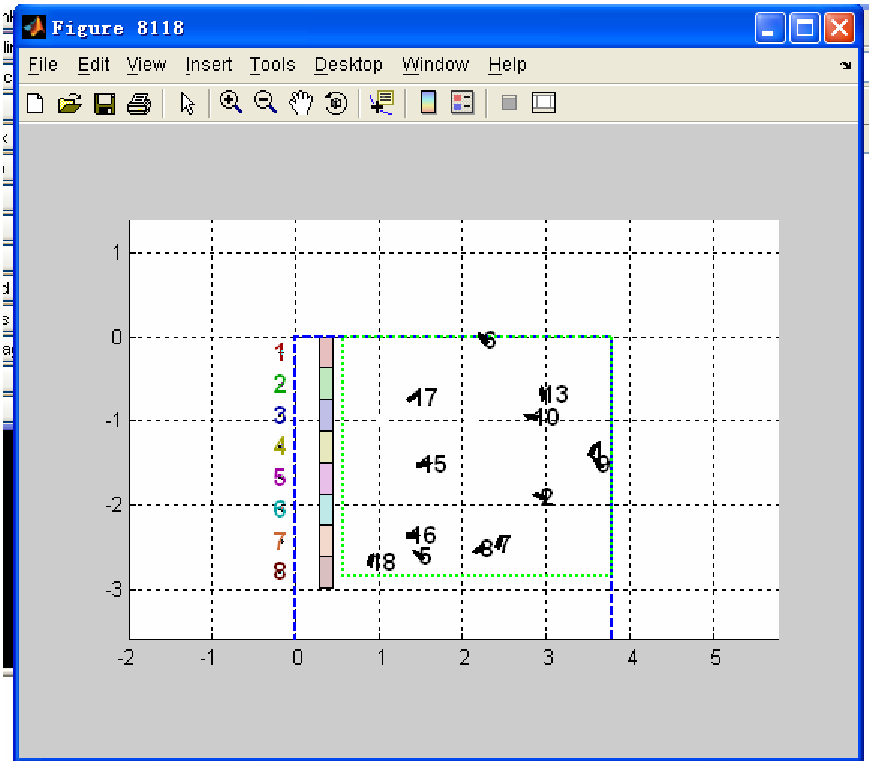

Figure 7.1 Low Resolution at Initialization Stage

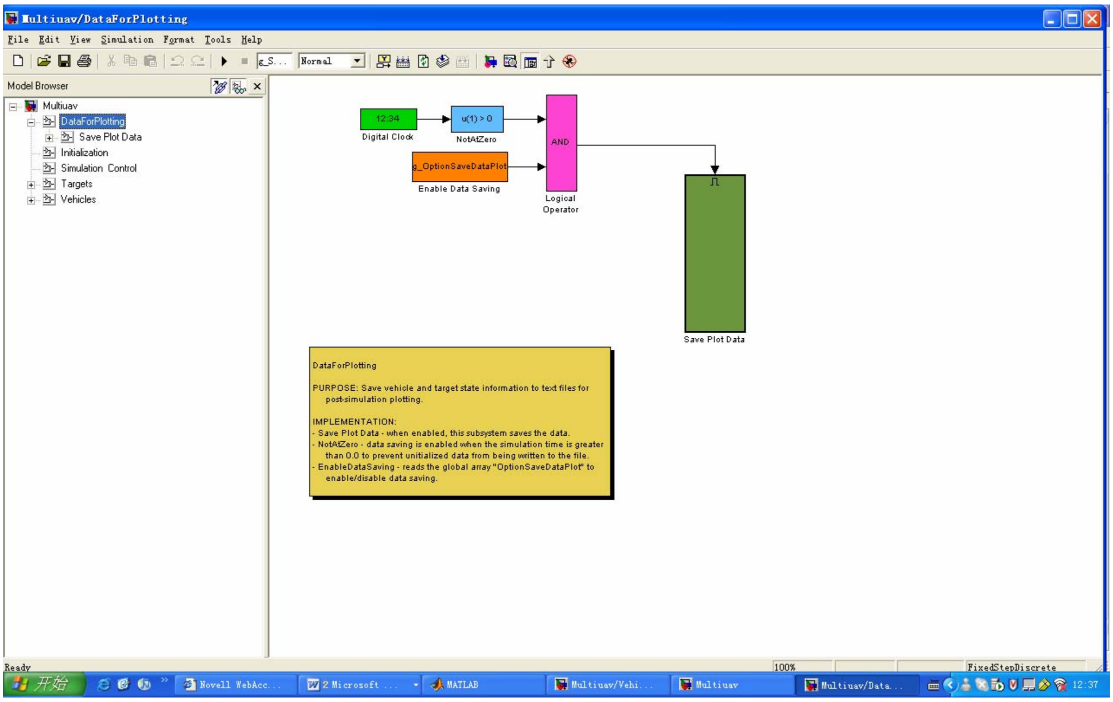

Figure 7.2 “Save Plot Data” Block in Simulink

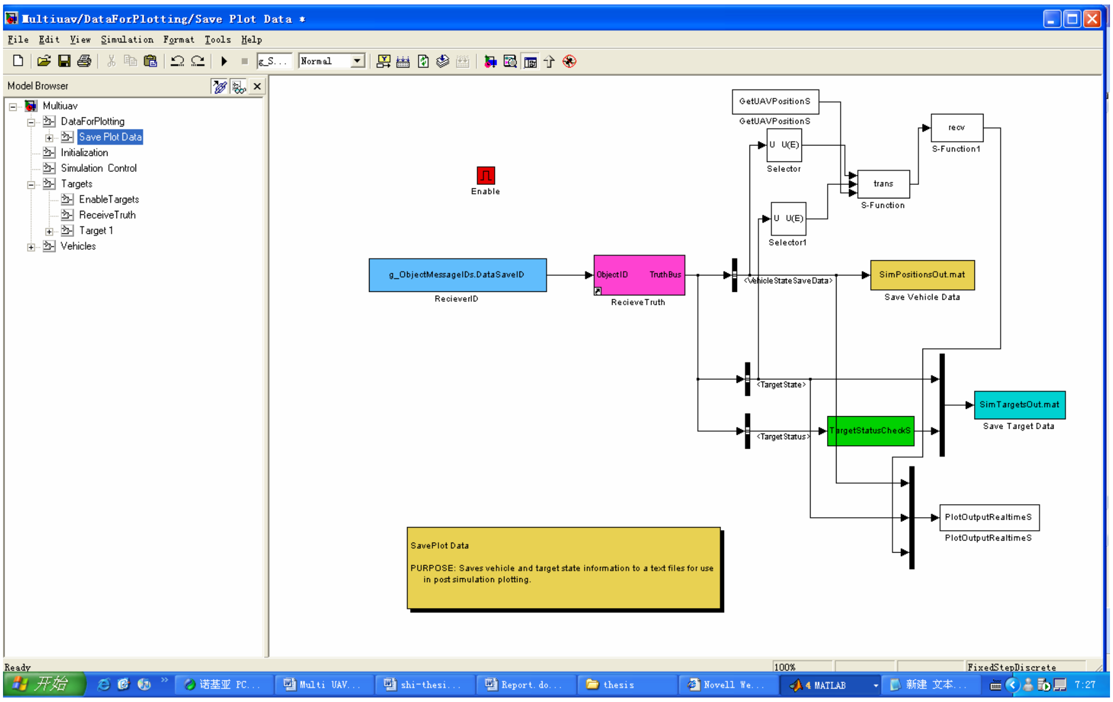

Figure 7.3 "Save Plot Data" inside Structure

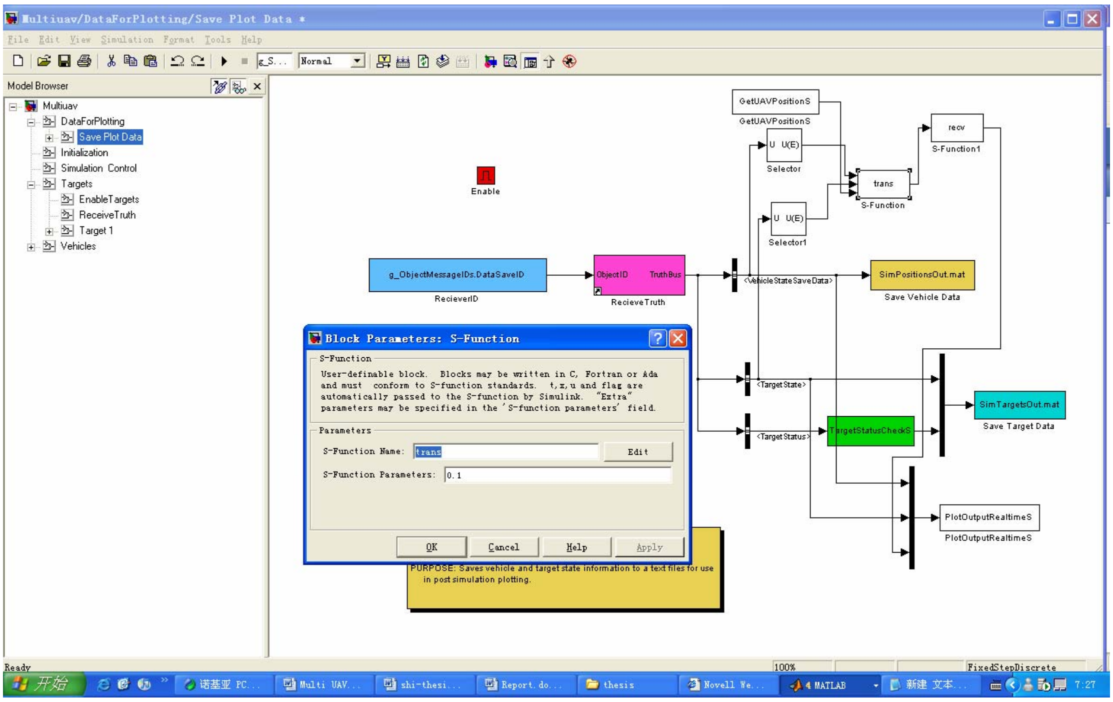

Figure 7.4 Time Unit of the S-Function is set to 0.1 second

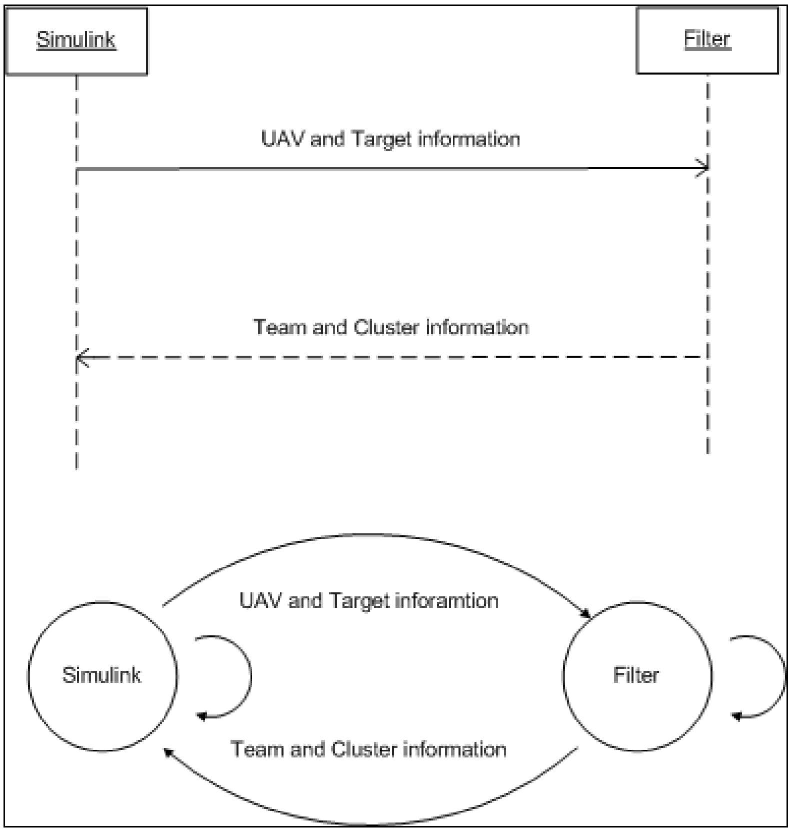

Figure 7.5 Simulink and Filter Communication Sequence Chart

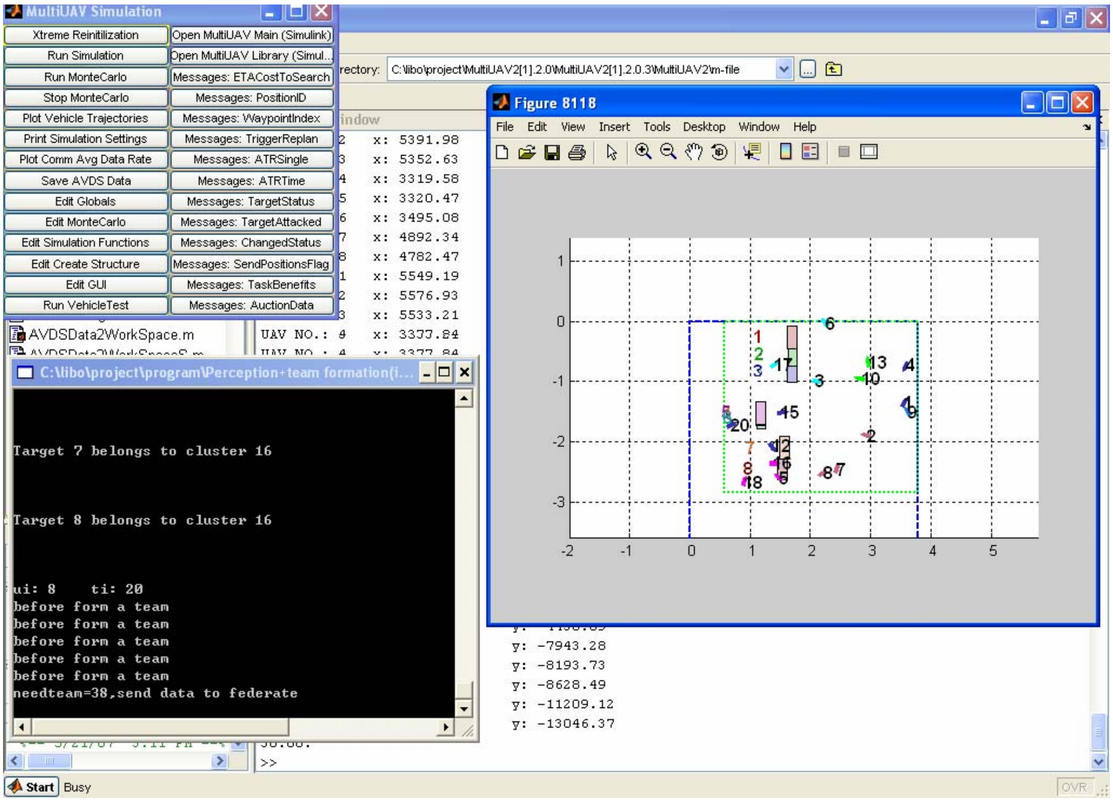

Figure 7.6 Communication between Simulink and Tactical Federate

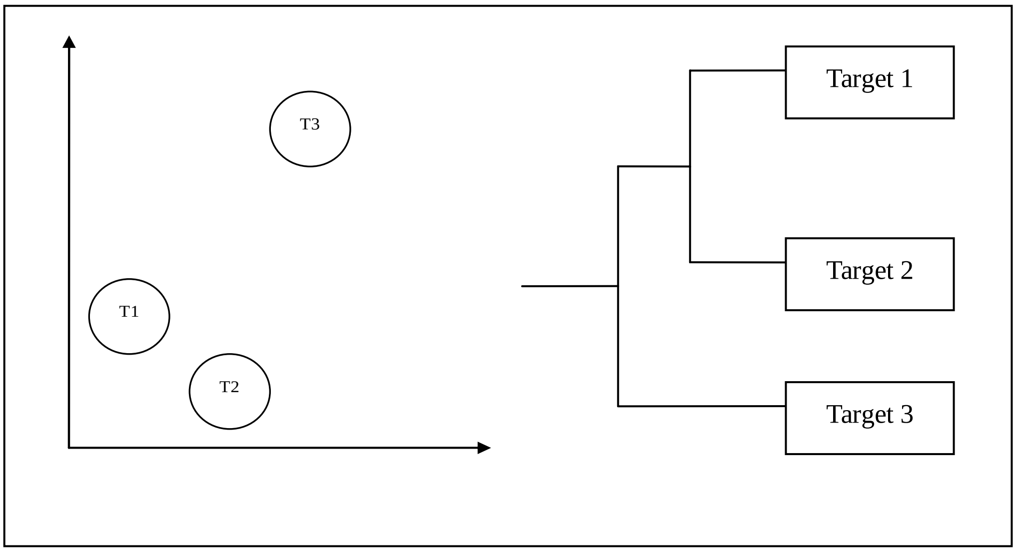

Figure 7.7 A Simple Targets Clustering Process

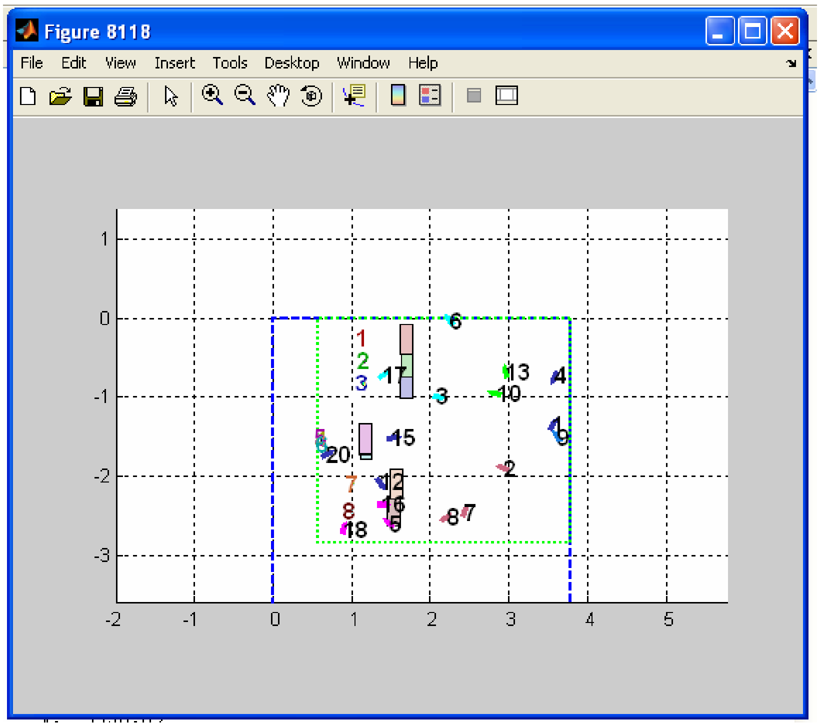

Figure 7.8 Targets are Clustered as Battalions

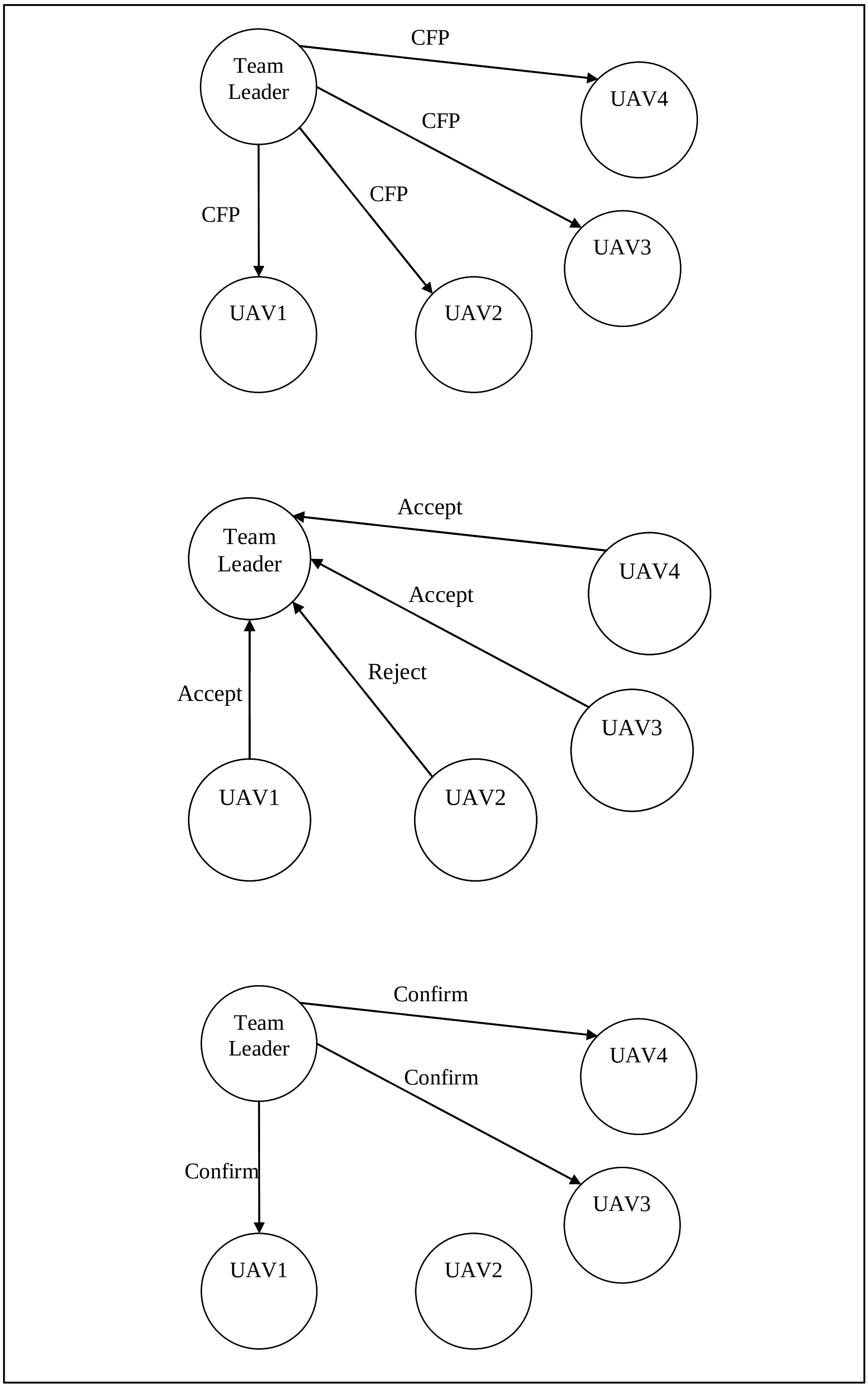

Figure 7.9 Team Formation Process Based on Contract Net

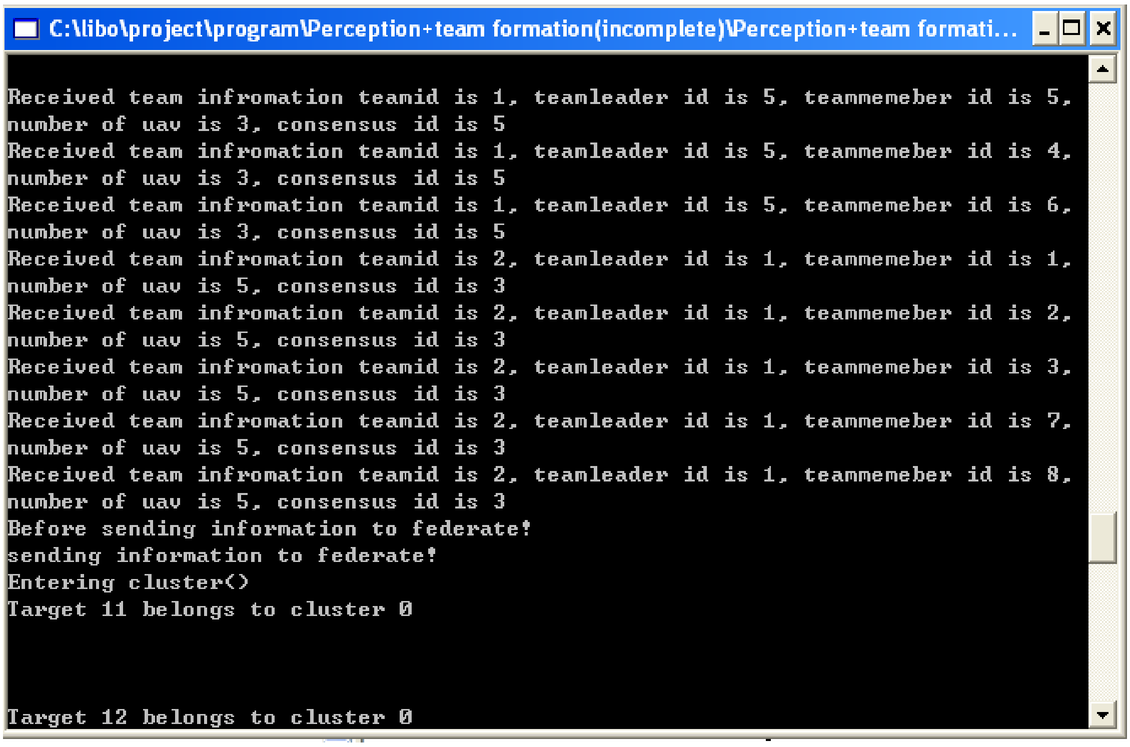

Figure 7.10 Team Formation Process Results

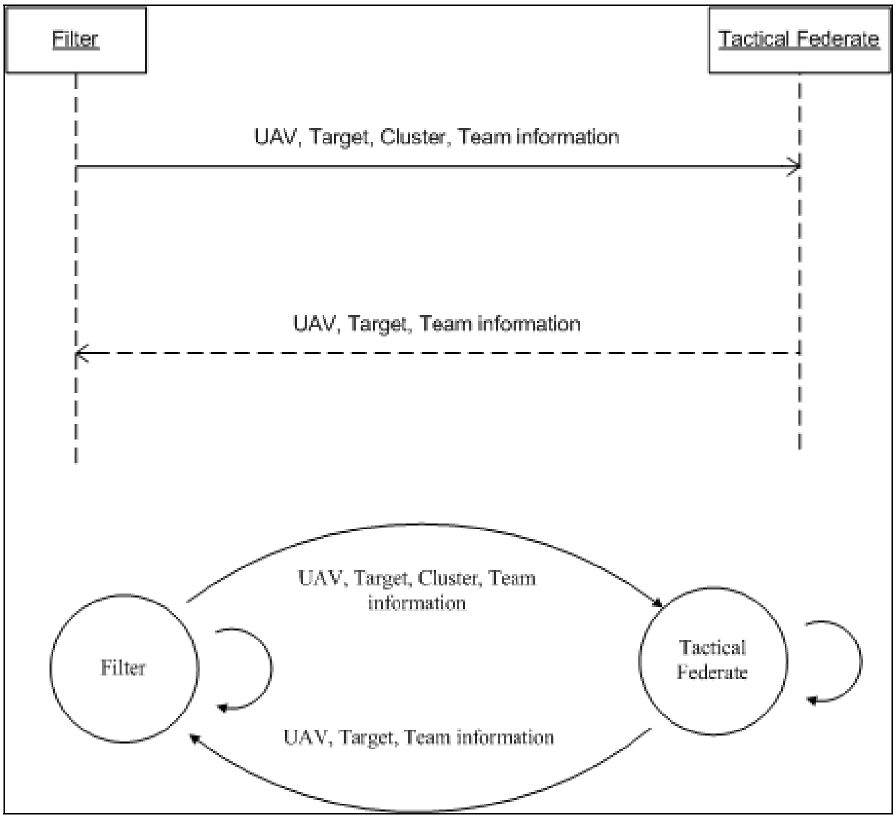

Figure 7.11 Filter and Tactical Federate Communication Sequence Chart

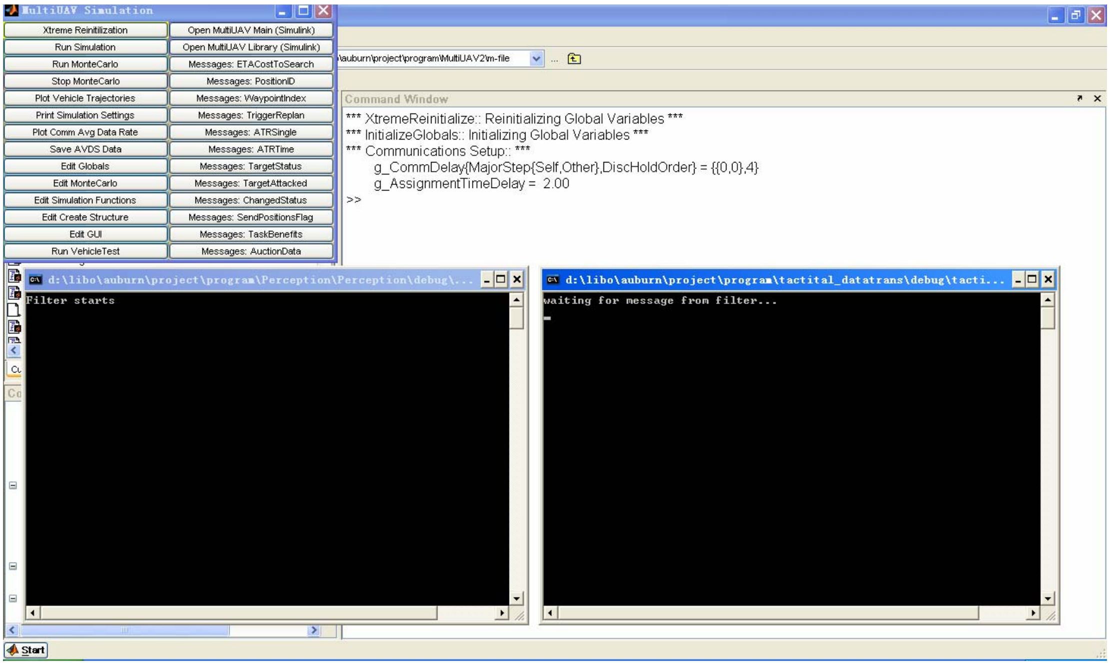

Figure 7.12 Communication between Filter and Tactical Federate

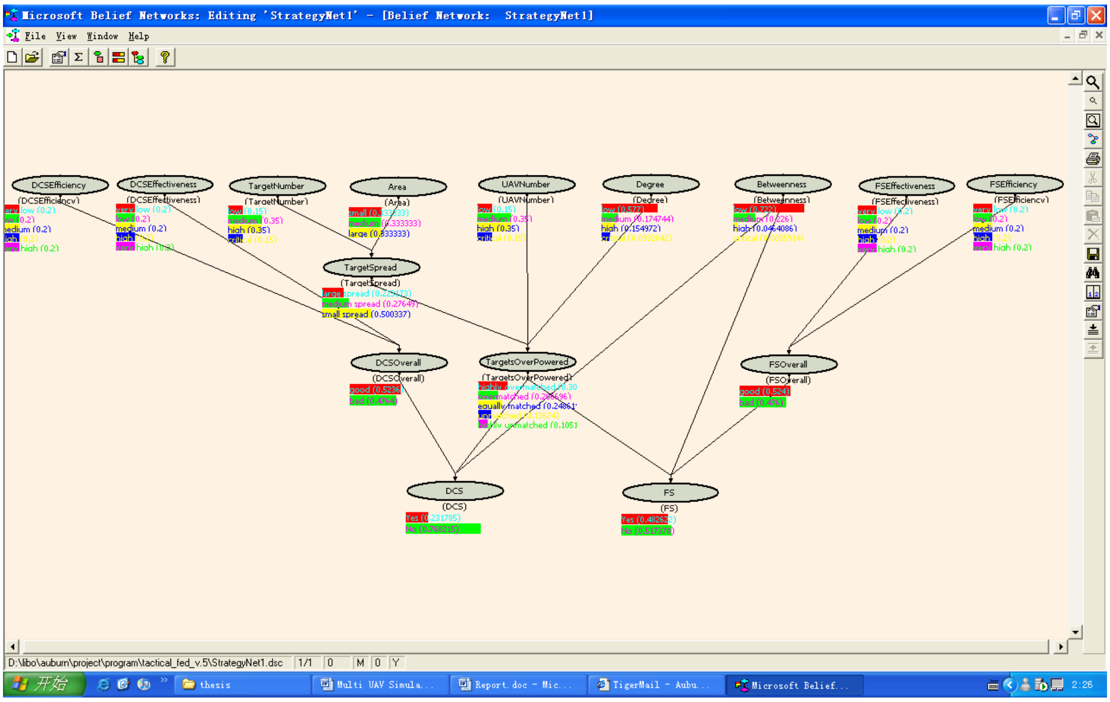

Figure 7.13 MSBNx Implementation

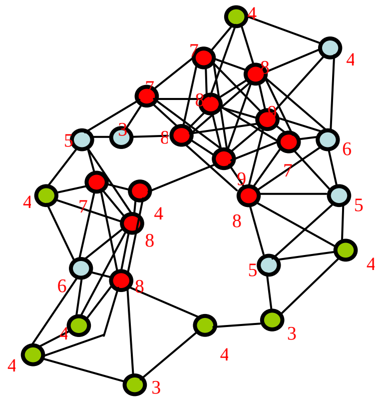

Figure 7.14 Sample of Critical Target

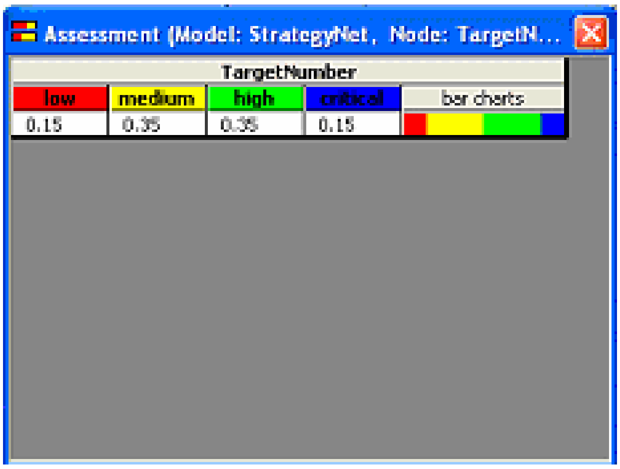

Figure 7.15 Assessment Table of “Number of Target” Node

Assessment (Model: StrategyNet, Node: DCS)

Parent Node(s)			DCS		
TargetsOverPowered	Betweenness	DCSOverall	Yes	No	bar charts
highly overmatched	low	good	0.361	0.639	
		bad	0.268	0.732	
	medium	good	0.588	0.412	
		bad	0.505	0.495	
	high	good	0.825	0.175	
		bad	0.722	0.278	
	critical	good	0.948	0.052	
		bad	0.856	0.144	
overmatched	low	good	0.258	0.742	
		bad	0.196	0.804	
	medium	good	0.454	0.546	
		bad	0.351	0.649	
	high	good	0.68	0.32	
		bad	0.577	0.423	
	critical	good	0.856	0.144	
		bad	0.794	0.206	
equally matched	low	good	0.124	0.876	
		bad	0.062	0.938	
	medium	good	0.278	0.722	
		bad	0.206	0.794	
	high	good	0.495	0.505	
		bad	0.412	0.588	
	critical	good	0.66	0.34	
		bad	0.557	0.443	
unmatched	low	good	0.072	0.928	
		bad	0.041	0.959	
	medium	good	0.144	0.856	
		bad	0.072	0.928	
	high	good	0.309	0.691	
		bad	0.196	0.804	
	critical	good	0.526	0.474	
		bad	0.433	0.567	
highly unmatched	low	good	0.031	0.969	
		bad	0.0	1.0	
	medium	good	0.124	0.876	
		bad	0.072	0.928	
	high	good	0.247	0.753	
		bad	0.155	0.845	
	critical	good	0.371	0.629	
		bad	0.268	0.732	

Figure 7.16 Assessment Table of DSC Node

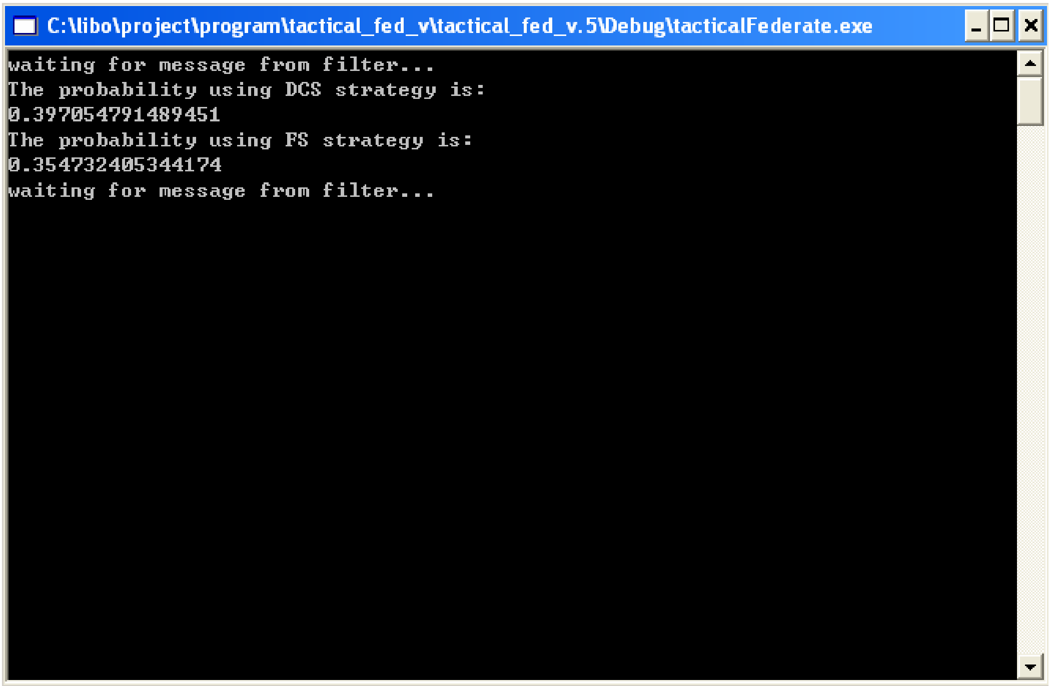

Figure 7.17 Calculation Results of Bayesian Network

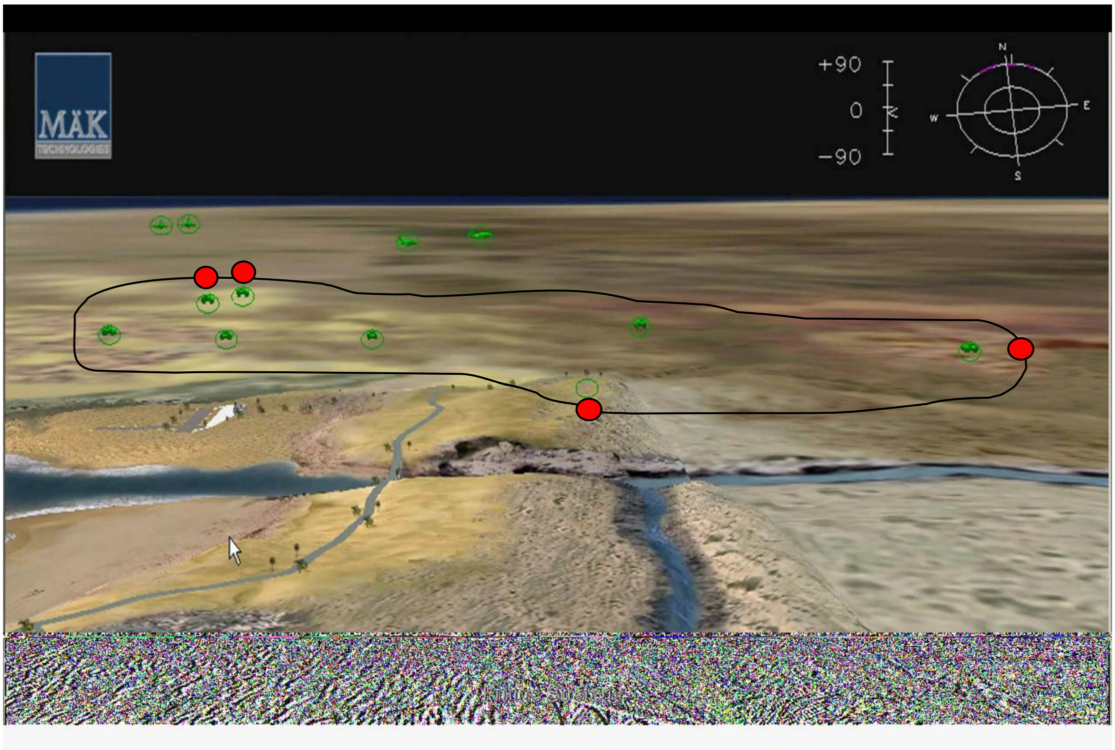

Figure 7.18 Simulation using Fringe Point strategy (1) - Fringe and Fringe Points of Targets Area

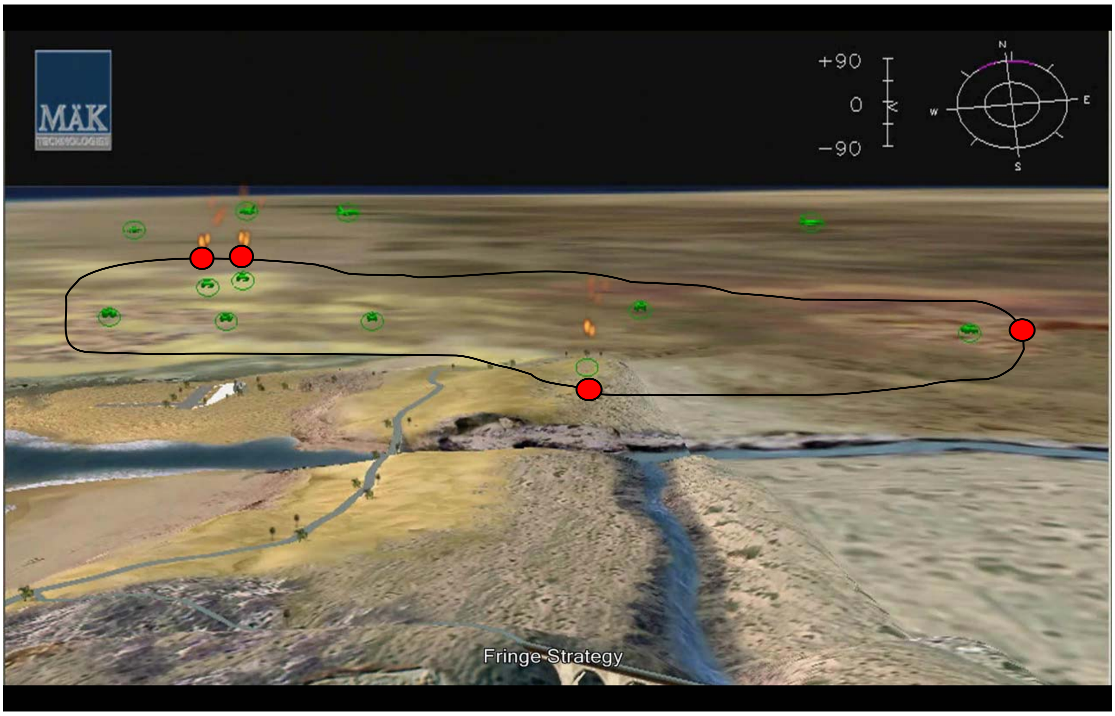

Figure 7.19 Simulation using Fringe Point Strategy (2)

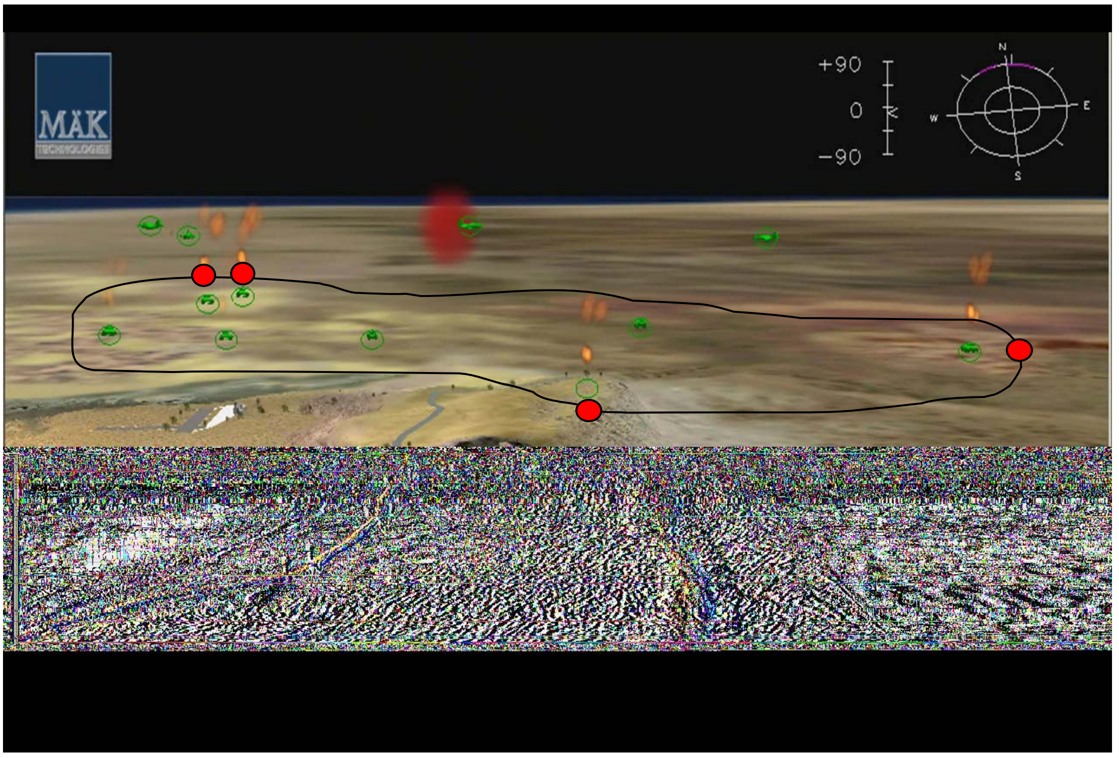

Figure 7.20 Simulation using Fringe Point Strategy (3)

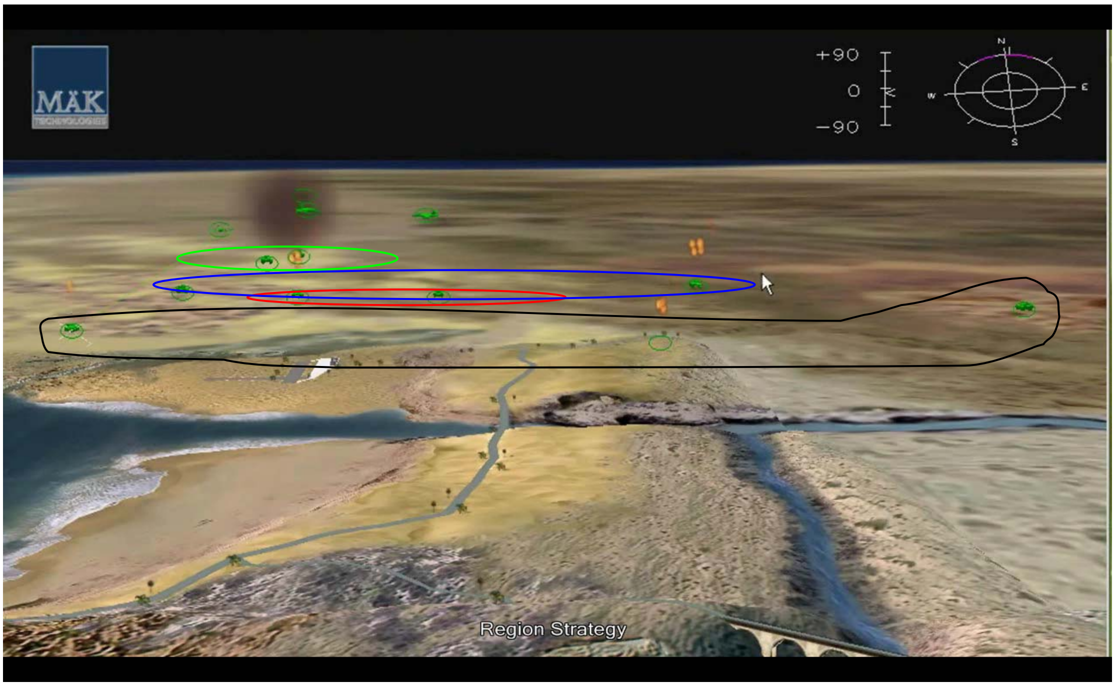

Figure 7.21 Simulation using Region Strategy

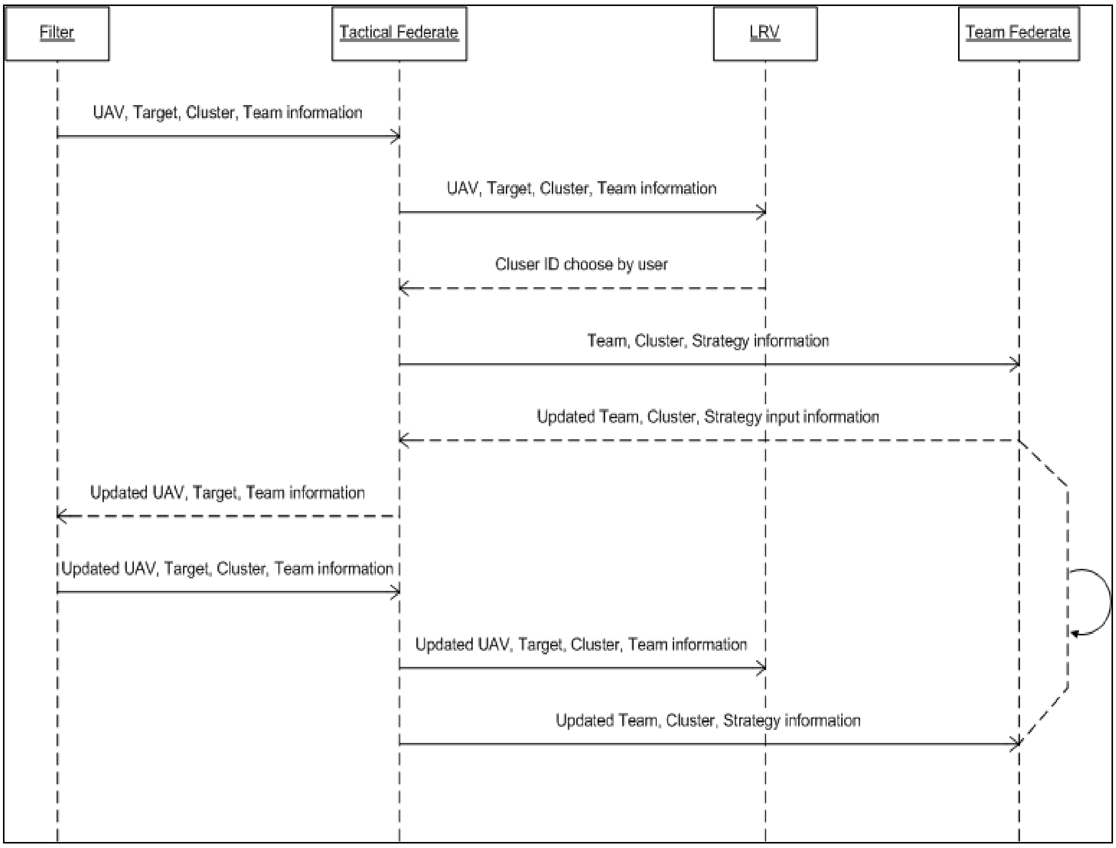

Figure 7.22 Program Sequence Chart of Tactical Federate Communication Functions

CHAPTER 8: SIMULATION RESULTS COMPARISON

In this simulation, we use two strategies, fringe point strategy and region strategy, to support UAV team attack battalion. The UAVs have different performance when they use the two strategies in the same situations.

Figure 8.1 shows the targets that are destroyed and the time lapse. The targets are destroyed by using region strategy is faster at the beginning and slower at the end than by using fringe point strategy. UAVs spend a little less time using fringe point strategy to destroy all the targets than using region strategy. Figure 8.2 shows the rate of UAV destroying targets using region strategy is much faster at the beginning and slower at the end compared to using fringe point strategy. The average target destroy rate of using fringe point strategy is 0.17 targets/second and the rate of using region strategy is 0.14 targets/second.

The reason of this is that using fringe point strategy, the UAVs firstly fly to the fringe points and then attack the closest target. While using region strategy, the targets areas are divided into four regions. Each UAV is assigned to attack one region at the beginning. The UAV do not need to fly to the fringe point and then attack, they perform in a region, a smaller area than the entire targets area. That makes UAVs spend less time to start to attack the targets. UAVs using fringe point strategy destroyed targets closest to the fringe and move to attack the targets in the inner region of the fringe. After about half of the simulation time, the rest of targets have smaller area compared to the initialized targets area. Because the updated fringe is smaller than before, it takes less time for UAV fly to the new fringe points to attack. Compared to the fringe point strategy, by using region strategy, some regions are very large and it takes some time for UAV fly from one target to another. In the smaller region, the UAVs could destroy all targets in a short time, but in the larger region, the UAVs spend a lot of time flying between the targets such as the region marked black in Figure 7.21. Figure 8.1 shows by using region strategy, the eighth target is destroyed at 44 seconds and the ninth target is destroyed at 66 seconds. Because the average target destroy rate by using region strategy is 0.14, in 20 seconds, there could be 2.8 targets to be destroyed, but in reality, there are no target destroyed during this time. The UAV that destroyed the last target, spent 41 seconds fly from the previous destroyed target to the last one. That causes the region strategy to spend more time to destroy all the targets than the fringe point strategy. The average target destroy rate using region strategy is smaller than that using the fringe point strategy. If the last target is not count in, the average target destroy rate using fringe point strategy is 0.19 and the rate using region strategy is 0.17. That means on the average, using fringe point strategy is a little faster at destroying target than using region strategy in this situation.

The results shows that UAVs using region strategy destroy target faster at the beginning than at the end; UAVs using fringe point strategy do not perform much differently. In Figure 8.1, using region strategy, the fourth target is destroyed at 13 second; using fringe point strategy, the fourth target is destroyed at 27 second. The average of target destroy rate of using fringe point strategy at 13 second is 0.17 targets/second; the rate of using region strategy is 0.31 targets/second. These results show that using region strategy has larger target destroy rate at the beginning of the simulation (13/66=20%, the simulation lasts 66 seconds).

The overall average of target destroy rate of using fringe point strategy is a bit larger than using region strategy. For the battlefield scenario, although using region strategy on the average is a little slower at destroying targets, the UAVs destroy targets faster at the beginning. That has some advantages for UAVs at the battlefield. UAVs destroy targets fast at the beginning could be safer for them than destroying them later. Because the targets in the real world could fire back to UAVs, destroying targets earlier could result in fewer threats to the UAVs and the UAVs will be safer for the rest of time. Compared to region strategy, using fringe point strategy has on the average a little larger target destroy rate and much smaller rate at the beginning of the simulation. UAV using this strategy are more dangerous. If the targets could fire back, there are larger probabilities for UAVs to be destroyed at the beginning than using region strategy. Suppose there are some UAVs destroyed at the beginning, there will be not as many UAVs to attack targets as before and it could take longer time for UAVs to fulfill the tasks and target destroy rate could dramatically drop down at the rest of time. In such a situation, it is not a good choice for UAVs to use fringe point strategy to attack the targets.

In this situation, using region strategy has more advantages and could fulfill the task with smaller number of UAVs destroyed compared to using fringe point strategy.

Efficiency is number of targets destroyed in a period of time. Effectiveness is defined as T/U. T is the number of targets destroyed in a period of time and U is the number of UAVs destroyed plus 1 in that period of time. That means if no UAV is destroyed at that period of time, effectiveness will be the number of targets destroyed at that period. Because the target cannot fight back and UAV is not destroyed. Effectiveness equals to efficiency. In this simulation, efficiency is shown in Figure 8.3. In the situation where targets could fire back, UAVs using fringe point strategy could have larger probabilities to be destroyed at the beginning. The efficiency of using the fringe point strategy could be smaller at beginning than it is shown in Figure 8.3. There could be less UAVs and many targets left after a while from beginning (a quarter of the total simulation time). Because UAVs destroy target faster using the region strategy than using fringe point strategy at the beginning, there will be more UAVs and fewer targets left after that period time. UAVs have larger probabilities not be destroyed, while UAVs using fringe point strategy are more vulnerable. The average efficiency of using region strategy could be larger than using fringe point strategy in this situation.

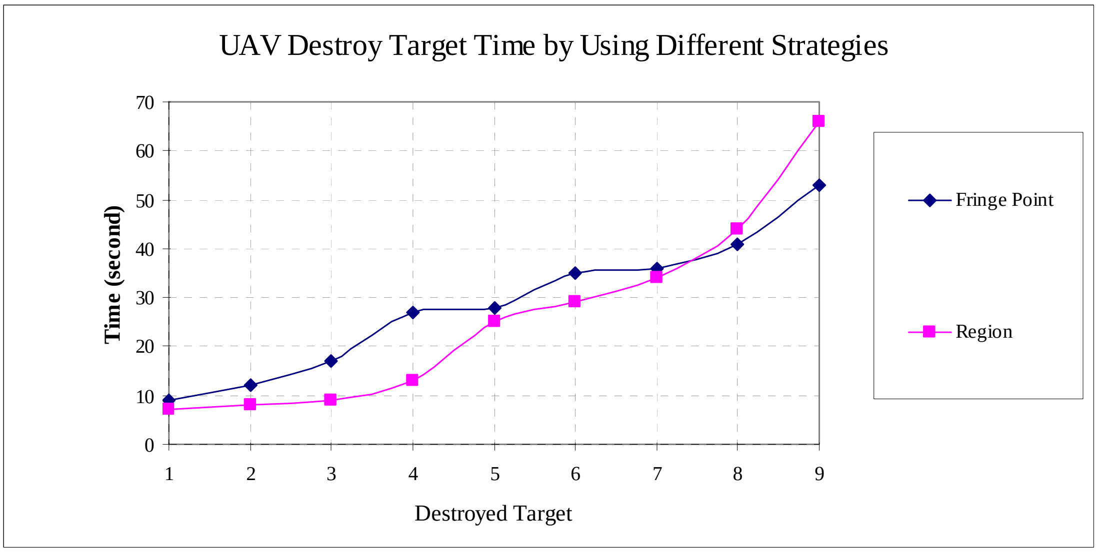

Figure 8.1 UAV Destroy Target by Time Lapse

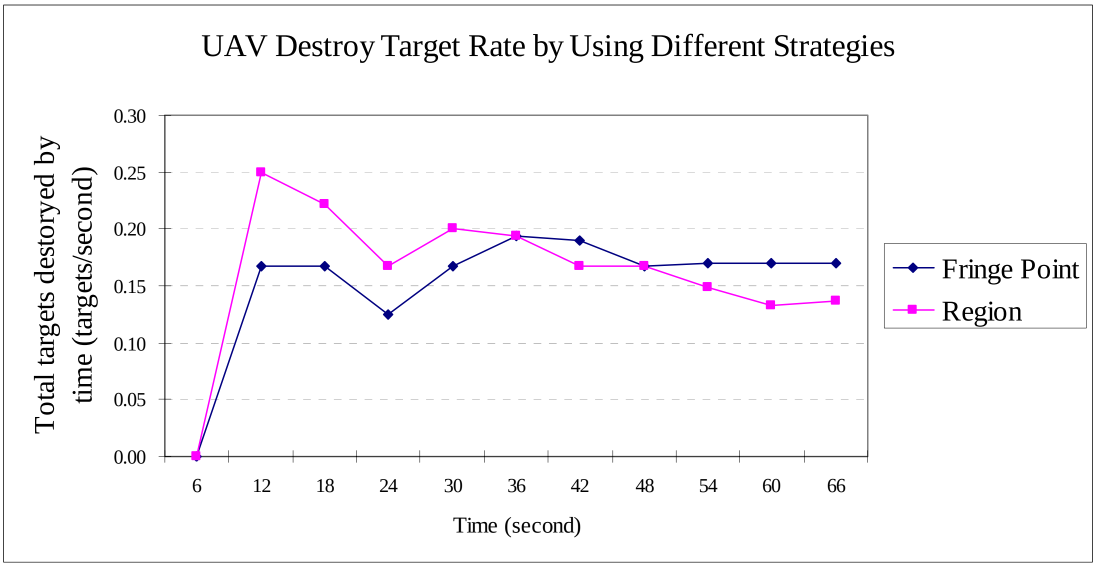

Figure 8.2 Target Destroy Rate by Using Different Strategies

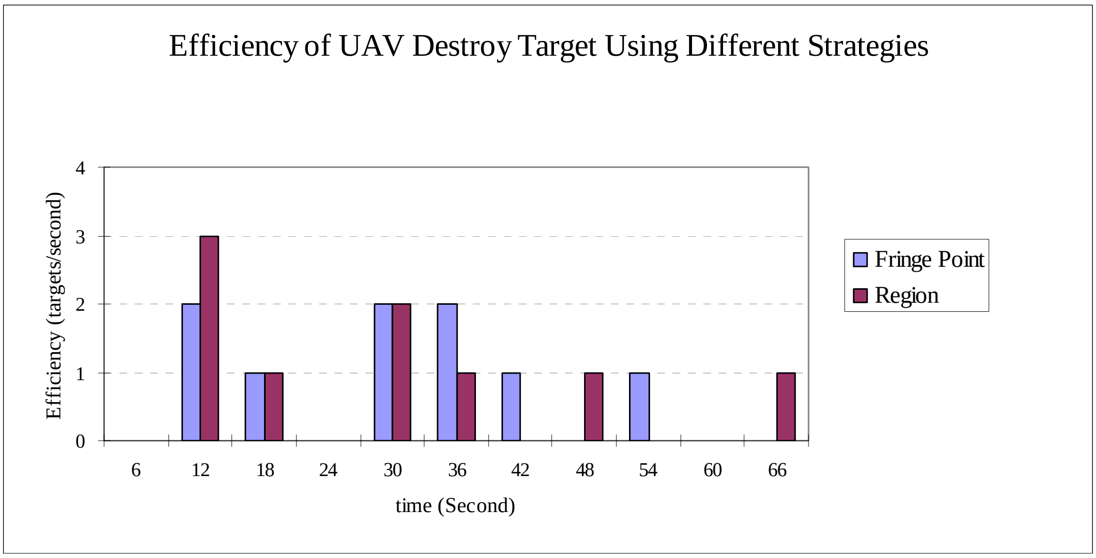

Figure 8.3 Efficiency of UAV Destroy Target Using the Strategies

CHAPTER 9: SUMMARY AND FUTURE WORKS

With the populating of using UAV in battlefield, multiple UAVs could work together to fulfill multiple tasks. Targets are clustered as battalions based on each UAV's perception. The UAV teams are implemented to attack battalions and have a consensus of perception within each team. The UAVs actions are important to make the attack efficient and effective. The decision making model using Bayesian Network to selects the proper strategy to support the UAV team in attacking in an efficient and effective manner in different situations. This simulation contains multisimulation, multiresolution and decision making models in multistage. Different parties work on different platforms and have a timely data exchange. The sub models can be evaluated, updated and upgraded dynamically at run-time. In this way, the consistencies between sub models and between resolutions are maintained. Multiple visualizations are utilized for both low resolution and high resolution. The animate and friendly multiple visualizations show the different stages of the resolutions of this simulation. That is convenient for the human controller and users to perceive the simulation.

Future works address the simulation of active targets. The targets could move and fire UAVs. That will make the simulation more like real world battlefield. Selecting a strategy for a team based on the evaluation of current battlefield by decision model and run-timely changing the strategy for the team will be simulated to achieve better effectiveness. Other issue will address interaction with human controller at run-time. The key of the decision control could be human control with decision model supports. Combination of human control and machine computation will ensure more accurate decision. The GUI of high resolution could be a tree structure, and each leaf of the tree represents a simulation among the multiple simulations. The human controller or users could pick up a leaf node on the tree to perceive a simulation such as a UAV team attacks a battalion using a specific strategy in the visualization.

REFERENCES

[1] Corner, J.J.; Lamont, G.B. Parallel simulation of UAV swarm scenarios. Simulation Conference, 2004. Proceedings of the 2004 Winter. Volume 1, Issue , 5-8 Dec. 2004 Page(s): - 363

[2] DoD. Modeling and Simulation (M&S) Glossary. DoD Publication 5000.59-M, Jan. 1998.

[3] P.K. Davis, and R. J.Hillestad. Experiments in Multiresolution Modeling. RAND Report. MR1004. RAND Corp.1998.

[4] Levent Yilmaz and Alvin Lim. Multisimulation with Multi-resolution, Multi-stage Models over Federated Simulation Infrastructures. October 2006.

[5] Brown, D. A. 1998. Medusa's Mirror: Stepping Forward to Look Back "Future UAV Design Implications From the 21st Century Battlefield". School of Advanced Military Studies; United States Army Command and General Staff College.

[6] Davis, C. R. 1995. Airborne reconnaissance: The leveraging tool for our future strategy. Executive Research Project, National Defense University. The Industrial College of the Armed Forces.

[7] McHale, J. 2003. Unmanned aircraft armed and dangerous. Military and Aerospace Electronics 14 (11): 18–25.

[8] French, M. 2003. Uavs advance since desert storm. Available online via <www.fcw.com/fcw/ articles/2003/0217/web-uav-02-19-03. asp>[accessed August 19, 2003].

[9] Reid G. Smith. The Contract Net Protocol: high-level communication and control in a distributed problem solver. IEEE TRANSACTIONS ON COMPUTERS, 1104 VOL. C-29, NO. 12, DECEMBER.

[10] J. B. MacQueen (1967): "Some Methods for classification and Analysis of Multivariate Observations", Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability, Berkeley, University of California Press, 1:281-297

[11] Lemaire, T.; Alami, R.; Lacroix, S.; A distributed tasks allocation scheme in multi-UAV context. Robotics and Automation, 2004. Proceedings. ICRA '04. 2004 IEEE International Conference on Volume 4, Apr 26-May 1, 2004 Page(s):3622 - 3627 Vol.4.

[12] Prithviraj Dasgupta. Distributed automatic target recognition using multi-agent UAV swarms. International Conference on Autonomous Agents. Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems. SESSION: Robotics, Pages: 479 – 481. 2006

[13] Sujit, P.B.; Ghose, D. Multiple UAV search using agent based negotiation scheme. American Control Conference, 2005. Proceedings of the 2005. Volume , Issue , 8-10 June 2005 Page(s): 2995 - 3000 vol. 5.

[14] Goktogan, A.H.; Nettleton, E.; Ridley, M.; Sukkarieh, S. Real time Multi-UAV Simulator. Robotics and Automation, 2003. Proceedings. ICRA apos;03. IEEE International Conference on Volume 2, Issue, 14-19 Sept. 2003 Page(s): 2720 - 2726 vol.2

[15] Joshua J. Corner, Gary B. Lamont. Parallel simulation of UAV swarm scenarios. Proceedings of the 36th conference on Winter simulation WSC '04. December 2004

[16] Song, Y.D.; Yao Li; Liao, X.H.; Orthogonal transformation based robust adaptive close formation control of multi-UAVs. American Control Conference, 2005. Proceedings of the 2005. 8-10 June 2005 Page(s):2983 - 2988 vol. 5.

[17] Dionne, Dany; Rabbath, Camille A.; Multi-UAV Decentralized Task Allocation with Intermittent Communications: the DTC algorithm. American Control Conference, 2007. ACC '079-13 July 2007 Page(s):5406 – 5411.

[18] Ahmadzadeh, A.; Jadbabaie, A.; Kumar, V.; Pappas, G.J.; Multi-UAV Cooperative Surveillance with Spatio-Temporal Specifications. Decision and Control, 2006 45th IEEE Conference on 13-15 Dec. 2006 Page(s):5293 – 5298.

[19] Kanchanavally, S.; Ordonez, R.; Schumacher, C.J.; Path planning in three dimensional environment using feedback linearization. American Control Conference, 2006. 14-16 June 2006 Page(s):6 pp.

[20] Paul F. Reynolds, Anand Natrajan, Sudhir Srinivasan. Consistency maintenance in multiresolution simulation. July 1997, ACM Transactions on Modeling and Computer Simulation (TOMACS), Volume 7 Issue 3.

[21] McLain, T.W.; Chandler, P.R.; Rasmussen, S.; Pachter, M.; Cooperative control of UAV rendezvous. American Control Conference, 2001. Proceedings of the 2001. Volume 3, 25-27 June 2001 Page(s):2309 - 2314 vol.3.

[22] Liu Baohong, Hao Jianguo, Huang Kedi. Parallel implementation of multi-resolution models in distributed simulation system. May 2007, Proceedings of the 18th conference on Proceedings of the 18th IASTED International Conference: modelling and simulation MOAS'07.

[23] K. S. Barber, N. Gujral, J. Ahn, D. DeAngelis, K. Fullam, D. C. Han, D. N. Lam, J. Park. Design, runtime, and analysis of multi-agent systems. July 2005, Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems AAMAS '05.

[24] Samin Karim, Clint Heinze. Experiences with the design and implementation of an agent-based autonomous UAV controller. Industrial track, aerospace applications. July 2005, Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems AAMAS '05.

[25] Myeong-Wuk Jang; Gul Agha. Scalable agent distribution mechanisms for large-scale UAV simulations. Integration of Knowledge Intensive Multi-Agent Systems, 2005. International Conference on Volume , Issue , April 18-21, 2005 Page(s): 85 - 90

[26] Ian C. Price, Gary B. Lamont. Military applications: military analysis I: GA directed self-organized search and attack UAV swarms. December 2006, Proceedings of the 38th conference on Winter simulation WSC '06.

[27] Gregory J. Barlow, Choong K. Oh. Robustness analysis of genetic programming controllers for unmanned aerial vehicles. Artificial life, evolutionary robotics, adaptive behavior. July 2006, Proceedings of the 8th annual conference on Genetic and evolutionary computation GECCO '06.

[28] Lamont, G.B.; Slear, J.N.; Melendez, K.; UAV Swarm Mission Planning and Routing using Multi-Objective Evolutionary Algorithms. Computational Intelligence in Multicriteria Decision Making, IEEE Symposium on 1-5 April 2007 Page(s):10 – 20.

[29] Liang Chen; Gregory, B.; Easley, M.; Peot, M.; Lee, J.; Altshuler, T.; Market-Based Collaborations for Autonomous Operations of Unmanned Air Vehicles. Distributed Intelligent Systems: Collective Intelligence and Its Applications, 2006. DIS 2006. IEEE Workshop on 15-16 June 2006 Page(s):273 – 278.

[30] Prithviraj Dasgupta, Matthew Hoeing. Market-Based Distributed Task Selection in Multi-agent Swarms. December 2006, Proceedings of the IEEE/WIC/ACM international conference on Intelligent Agent Technology IAT '06.

[31] Yanli Yang; Minai, A.A.; Polycarpou, M.M.; Evidential map-building approaches for multi-UAV cooperative search. American Control Conference, 2005. Proceedings of the 2005 June 8-10, 2005 Page(s):116 – 12.

[32] Huahui Yi; Bifeng Song; Dongchao Ji; Tianxiang Yu. Experiment Research on Situation Awareness of the Operators for Unmanned Aerial Vehicle. Computational Intelligence and Security, 2006 International Conference. 3-6 Nov. 2006. Volume: 2, On page(s): 1225-1228

[33] Suresh Jeyaraman; Tsourdos, A.; Ratal Zbikowski; White, B.; Formal techniques for the modelling and validation of a co-operating UAV team that uses Dubins set for path planning. American Control Conference, 2005. Proceedings of the 2005. 8-10 June 2005. On page(s): 4690 - 4695 vol. 7

[34] David Heckerman; A Tutorial on Learning with Bayesian Network. Microsoft research, Advanced technology division, 1995. MSR-TR-95-06.

[35] Gowtham, G.; Kumar, K.S.; Simulation of multi UAV flight formation. Digital Avionics Systems Conference, 2005. DASC 2005. The 24th Volume 2, 30 Oct.-3 Nov. 2005 Page(s):6 pp. Vol. 2.

[36] Ryan, A.; Zennaro, M.; Howell, A.; Sengupta, R.; Hedrick, J.K.; An overview of emerging results in cooperative UAV control. Decision and Control, 2004. CDC. 43rd IEEE Conference on Volume 1, 14-17 Dec. 2004 Page(s):602 - 607 Vol.1

[37] Zhongjian Li; Jinwen An; Dong Xu; HIL Simulation of High Altitude Long Endurance UAV's Target Tracking/Flight Integrated Control. Intelligent Control and Automation, 2006. WCICA 2006. The Sixth World Congress on Volume 2, 21-23 June 2006 Page(s):6188 – 6192.

[38] Edrich, M.; Schmalenberger, R.; Combined DSSS/FHSS approach to interference rejection and navigation support in UAV communications and control. Spread Spectrum Techniques and Applications, 2002 IEEE Seventh International Symposium on Volume 3, 2002 Page(s):687 - 691 vol.3.

[39] Borrelli, F.; Subramanian, D.; Raghunathan, A.U.; Biegler, L.T.; MILP and NLP Techniques for centralized trajectory planning of multiple unmanned air vehicles. American Control Conference, 2006 14-16 June 2006 Page(s):6 pp.

[40] Augeri, C.; Neebel, D.; Baird, L.; de Freitas, A.; UAV Communications: Integrating a Real-World Scenario with Computer Architecture. Frontiers in Education, 2005. FIE '05. Proceedings 35th Annual Conference 19-22 Oct. 2005 Page(s):F2H-15 - F2H-20

[41] Si, Yang; Xiaomin, Li; Hui, Xie. UAV Servicing and Training System Based on VR and HLA. Electronic Measurement and Instruments, 2007. ICEMI '07. 8th International Conference on Aug. 16 2007-July 18 2007 Page(s):2-341 - 2-345

[42] Ye, Z.; Bhattacharya, P.; Mohamadian, H.; Majlesein, H.; Yongmao Ye; Equational dynamic modeling and adaptive control of UAV. System of Systems Engineering, 2006 IEEE/SMC International Conference on 24-26 April 2006 Page(s):5 pp.

[43] Jiecai Luo; Some new optimal control problems in UAV cooperative control with information flow constraints. American Control Conference, 2003. Proceedings of the 2003 Volume 3, 4-6 June 2003 Page(s):2181 - 2186 vol.3.

[44] Dongli, Yuan; Jianguo, Yan; Xinmin, Wang; Qingbiao, Xi; A Study of Information Fusion for UAV Based on RBF Neural Network. Control and Automation, 2007. ICCA 2007. IEEE International Conference on May 30 2007-June 1 2007 Page(s):2839 – 2842.

[45] Cook, Kendra L. B.; The Silent Force Multiplier: The History and Role of UAVs in Warfare. Aerospace Conference, 2007 IEEE on 3-10 March 2007. Page(s):1 – 7.

[46] Wigley, G.; Jasiunas, M.; A low cost, high performance reconfigurable computing based unmanned aerial vehicle. Aerospace Conference, 2006 IEEE 4-11 March 2006 Page(s):13 pp.

[47] Mehdi Alighanbari; How, J.P.; Cooperative task assignment of unmanned aerial vehicles in adversarial environments. American Control Conference, 2005. Proceedings of the 2005. 8-10 June 2005 Page(s):4661 - 4666 vol. 7

[48] Shima, T.; Rasmussen, S.J.; Chandler, P.; UAV team decision and control using efficient collaborative estimation. American Control Conference, 2005. Proceedings of the 2005 on 8-10 June 2005 Page(s):4107 - 4112 vol. 6

[49] Nikolos, I.K.; Valavanis, K.P.; Tsourveloudis, N.C.; Kostaras, A.N.; Evolutionary algorithm based offline/online path planner for UAV navigation. Systems, Man and Cybernetics, Part B, IEEE Transactions on Volume 33, Issue 6, Dec. 2003 Page(s):898 – 912

[50] Fengqing, Zhang; Quancun, Kong; Distributed Remote Control System of UAV Based on Man-in-loop Real-time Operation. Control Conference, 2007. CCC 2007. Chinese July 26 2007-June 31 2007 Page(s):119 - 122

[51] Yun-an Hu; Bin Zuo; Xiaodong Li; The Application of an Annealing Recurrent Neural Network for Extremum Seeking Algorithm to Optimize UAV Tight Formation Flight. Computational Engineering in Systems Applications, IMACS Multiconference on Volume 1, 4-6 Oct. 2006 Page(s):613 - 620

[52] Sousa, J.; Simsek, T.; Varaiya, P.; Task planning and execution for UAV teams. Decision and Control, 2004. CDC. 43rd IEEE Conference on Volume 4, 14-17 Dec. 2004 Page(s):3804 - 3810 Vol.4.

[53] Ambrosino, G.; Ariola, M.; Ciniglio, U.; Corraro, F.; Pironti, A.; Virgilio, M.; Algorithms for 3D UAV Path Generation and Tracking. Decision and Control, 2006 45th IEEE Conference on 13-15 Dec. 2006 Page(s):5275 – 5280.

[54] Lundell, M.; Tang, J.; Nygard, K.; Fuzzy Petri net for UAV decision making. Collaborative Technologies and Systems, 2005. Proceedings of the 2005 International Symposium on 15-20 May 2005 Page(s):347 – 352.

[55] Yan Jin; Minai, A.A.; Polycarpou, M.M.; Cooperative real-time search and task allocation in UAV teams. Decision and Control, 2003. Proceedings. 42nd IEEE Conference on Volume 1, 9-12 Dec. 2003 Page(s):7 - 12 Vol.1.

[56] A Tutorial on Clustering Algorithms.
http://home.dei.polimi.it/matteucc/Clustering/tutorial_html/kmeans.html